

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平11-213179

(43)公開日 平成11年(1999) 8月6日

(51)Int.Cl.⁶

識別記号

F I

G 0 6 T 17/00

G 0 6 F 15/62

3 5 0 A

A 6 3 F 9/22

A 6 3 F 9/22

G

G 0 6 F 13/00

3 5 5

G 0 6 F 13/00

3 5 5

審査請求 未請求 請求項の数3 F D (全 38 頁)

(21)出願番号 特願平10-26409

(22)出願日 平成10年(1998) 1月23日

(71)出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72)発明者 松田 晃一

東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内

(72)発明者 内藤 剛人

東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内

(72)発明者 上野 比呂至

東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内

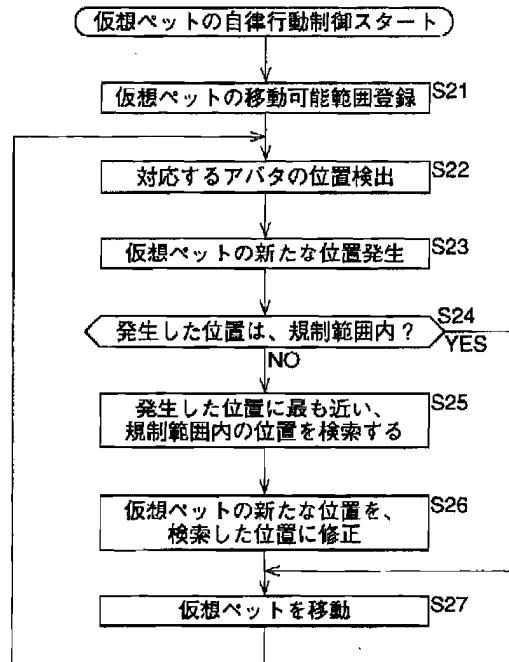
(74)代理人 弁理士 稲本 義雄

(54)【発明の名称】 共有仮想空間提供装置および方法、並びに提供媒体

(57)【要約】

【課題】 障害部との衝突を検知することなく、実世界では有り得ない行動を回避する。

【解決手段】 ステップS21において、仮想ペットが移動できる範囲を規定するデータが、例えば、HHD13aの中の移動範囲管理テーブルに登録される。次に、ステップS22において、仮想ペットに対応するアバタの位置が検出され、ステップS23においては、検出されたアバタの位置に基づいて、仮想ペットの位置が発生される。次に、ステップS24において、ステップS23で発生した仮想ペットの位置が、仮想ペットが移動できる規制範囲内であるか否かが判定される。ステップS23で発生した仮想ペットの位置が、規制範囲外であると判定された場合、ステップS25に進み、発生した仮想ペットの位置に最も近い規制範囲内の位置が検索される。次に、ステップS26において、仮想ペットの位置が、ステップS25で検索された位置に修正される。次に、ステップS27において、仮想ペットは、さらに、アバタに追従して移動する。



【特許請求の範囲】

【請求項 1】 ネットワークを介して接続されている複数のクライアント装置に共有仮想空間を提供する共有仮想空間提供装置において、

前記共有仮想空間内に存在する仮想生命オブジェクトの移動範囲を規定するデータを記憶する記憶手段と、

前記仮想生命オブジェクトの前記共有仮想空間における新たな位置のデータを発生する発生手段と、

発生された位置を前記規定されている移動範囲内の位置に修正する修正手段と、

前記修正された位置に前記仮想生命オブジェクトを移動させる移動手段とを備えることを特徴とする共有仮想空間装置。

【請求項 2】 ネットワークを介して接続されている複数のクライアント装置に共有仮想空間を提供する共有仮想空間提供装置において、

前記共有仮想空間内に存在する仮想生命オブジェクトの移動範囲を規定するデータを記憶する記憶ステップと、

前記仮想生命オブジェクトの前記共有仮想空間における新たな位置のデータを発生する発生ステップと、

発生された位置を前記規定されている移動範囲内の位置に修正する修正ステップと、

前記修正された位置に前記仮想生命オブジェクトを移動させる移動ステップとを備えることを特徴とする共有仮想空間提供方法。

【請求項 3】 ネットワークを介して接続されている複数のクライアント装置に共有仮想空間を提供する共有仮想空間提供装置に用いられるコンピュータプログラムにおいて、

前記共有仮想空間内に存在する仮想生命オブジェクトの移動範囲を規定するデータを記憶する記憶ステップと、

前記仮想生命オブジェクトの前記共有仮想空間における新たな位置のデータを発生する発生ステップと、

発生された位置を前記規定されている移動範囲内の位置に修正する修正ステップと、

前記修正された位置に前記仮想生命オブジェクトを移動させる移動ステップとを備えるコンピュータプログラムを提供することを特徴とする提供媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、共有仮想空間提供装置および方法、並びに提供媒体に関し、特に、例えば、3次元共有仮想空間において、仮想生命オブジェクトが、障害物との衝突を検知する必要なく、実世界では有り得ない行動を回避するようにした共有仮想空間提供装置および方法、並びに提供媒体に関するものである。

【0002】

【従来の技術】 従来、NIFTY-Serve(商標)や米国のCompu Serve(商標)などのように、複数の利用者が自らのパーソナルコンピュータを、モデムおよび公衆電話回線網を

介してセンターのホストコンピュータに接続し、所定の通信プロトコルに基づいてホストコンピュータにアクセスする、いわゆるパソコン通信サービスの分野においては、Habitat(商標)と呼ばれるサイバースペースのサービスが知られている。

【0003】 Habitatは、米国LucasFilm社によって1985年から開発が始められ、米国商業ネットワークであるQuantumLink社で約3年間運用された後、1990年2月に富士通Habitat(商標)としてNIFTY-Serveでそのサービスが開始されたものである。このHabitatにおいては、2次元グラフィックスで描画された「ポピュロポリス(Popolopolis)」と呼ばれる仮想の都市に、アバタ(avatar; インド神話に登場する神の化身)と呼ばれるユーザの分身を送り込み、ユーザ同士がチャット(Chat; 文字の入力と表示によるテキストベースでのリアルタイムの対話)などを行うことができる。このHabitatの更に詳細な説明については、サイバースペース、マイケル・ベネディクト編、1994年3月20日初版発行、NTT出版 ISBN 4-87188-265-9C0010(原著; Cyberspace: First Steps, Michael Benedikt, ed. 1991, MIT Press Cambridge, MA ISBN 0-262-02327-X) 第282頁乃至第307頁を参照されたい。

【0004】 この種のパソコン通信サービスで運用されている従来のサイバースペースシステムにおいては、仮想的な街並みや部屋の内部の様子が2次元グラフィックスで描画されており、アバタを奥行きまたは手前方向へ移動させる場合、単にアバタを2次元グラフィックスの背景上で上下に移動させるだけであり、仮想空間内での歩行や移動を疑似体験させるには表示の上での表現力が乏しかった。また、自分の分身であるアバタと他人のアバタが表示された仮想空間を、第3者の視点で見ることになるため、この点においても、疑似体験の感覚が損なわれるものであった。

【0005】 そこで、特開平9-81781号公報に開示されているように、仮想空間を3次元グラフィックスで表示し、ユーザがアバタの視点で自由に歩き回れる機能が、VRML(Virtual Reality Modeling Language)と呼ばれる3次元グラフィックス・データの記述言語を利用することによって実現されている。また、ユーザの代理となるアバタを用いてチャットを行う種々のサイバースペースの考察に関しては、日経エレクトロニクス1996.9.9(no. 670)の第151頁乃至159頁に記載されている。

【0006】 一方、近年、パーソナルコンピュータ用のソフトウェア・プログラムとして、熱帯魚を育てる飼育シミュレーション・ゲームや、仮想世界に住んでいる人工知能をもった仮想生物を育てる飼育シミュレーション・ゲームなどが知られている。また、電子手帳に犬や猫等の疑似ペットを表示して、その成長過程を楽しめるようにした製品も知られている(日経エレクトロニクス1997.4.7(no. 686)の第131頁乃至134頁参照)。さらには、

この種の飼育シミュレーション・ゲーム・プログラムを組込んだ、たまご大の携帯電子ペットとして、バンダイ株式会社が開発・製品化した「たまごっち（商標）」が広く知られている。

【0007】この種の携帯電子ペットは、1チップのLSI（大規模集積回路）にCPU（中央処理装置）やROM、RAM等が内蔵されており、そのROM内に飼育シミュレーション・ゲーム・プログラムが格納され、さらにLCD（液晶表示装置）によって仮想的なペットの姿や状態が表示されるようになっている。ユーザは、操作ボタンを操作して、「食事を与える」、「掃除をする」など、ペットとしての仮想生物を飼育するのに必要な指示を与える。この結果として、LCDで表示される仮想生物が成長し、その成長の過程で、仮想生物の外観が、例えば、たまご、ひよこ、成鳥へと段階的に変化していく。

【0008】また、ユーザのボタン操作に応じて、その指示が適切ならば順調に仮想生物が成長し、不適切ならば、病気になったり死亡してしまうようにプログラムされている。さらに、内部に組込まれているカレンダー・タイマーによって得られる仮想生物の誕生時点からの経過時間に基づいて、仮想生物側より各種の要求が行われるようにプログラムされている。例えば夜間の時間帯では仮想生物より睡眠の要求がなされ、食事の時間帯には食事の要求があり、またランダムにおやつや遊びの要求がなされる。これらの要求にユーザが適切に対応しない場合、仮想生物の成長が遅れたり、性格が悪化したりする。一方、ユーザが適切に対応した場合には、仮想生物の寿命が伸びるようにプログラムされている。

【0009】ここで、例えば、特開平07-160853号公報には、電子手帳などに適用され、動物や植物等の生物の成長過程に応じた画像を表示する技術が開示されている。すなわち、植物キャラクタの成長過程の各段階のビットマップ画像をROMに格納しておき、成長度に応じた植物キャラクタをLCDに表示させると共に、予めROMに記憶されている植物成長要素（水、光、肥料）の各キャラクタを表示させ、それら各成長要素の投与量をキー入力することで、その投与量に応じた各成長要素の値がRAM内の水量レジスタ、光量レジスタ、肥料量レジスタに各々セットされ、これらの各レジスタの値に基づいて、新たな成長度が算出され、その算出された成長度に対応した植物キャラクタがROMから読み出されてLCDに表示される。これにより、ユーザの飼育状況に応じた植物の成長過程が表示される。

【0010】

【発明が解決しようとする課題】以上のように、疑似ペット（以下、仮想ペット、または仮想生命オブジェクトとも記述する）は、飼い主であるユーザの操作に応じて様々な行動をとる。ところで、従来の共有仮想空間において、仮想ペットは、現世界では有り得ない行動、例えば、障害物（例えば、家の壁）に衝突し、そのままその

障害物を通り抜けてしまうような行動を回避するために、仮想ペットは、障害物との衝突または衝突しそうになることを検知し、障害物との衝突を回避し、そして、現世界で有り得る行動を促す移動経路（例えば、家の壁を通り抜けたらしないような経路）を探し出し、その経路を移動していた。しかしながら、仮想ペットが、移動しながら障害物との衝突を検知するためには、仮想ペットの行動を制御する複雑なプログラムが必要になるばかりか、そのプログラム起動されるCPUに付加がかかり、その結果、CPUの迅速な処理が妨げられる課題があった。

【0011】本発明はこのような状況に鑑てなされたものであり、仮想ペットが移動できる範囲を予め設定しておくことによって、仮想ペットが障害物との衝突を検知することなく、現世界では有り得ない行動を回避するようにするものである。

【0012】

【課題を解決するための手段】請求項1に記載の共有仮想空間提供装置は、共有仮想空間内に存在する仮想生命オブジェクトの移動範囲を規定するデータを記憶する記憶手段と、仮想生命オブジェクトの共有仮想空間における新たな位置のデータを発生する発生手段と、発生された位置を規定されている移動範囲内の位置に修正する修正手段と、修正された位置に仮想生命オブジェクトを移動させる移動手段とを備えることを特徴とする。

【0013】請求項2に記載の共有仮想空間提供方法は、共有仮想空間内に存在する仮想生命オブジェクトの移動範囲を規定するデータを記憶する記憶ステップと、仮想生命オブジェクトの共有仮想空間における新たな位置のデータを発生する発生ステップと、発生された位置を規定されている移動範囲内の位置に修正する修正ステップと、修正された位置に仮想生命オブジェクトを移動させる移動ステップとを備えることを特徴とする。

【0014】請求項3に記載の提供媒体は、共有仮想空間内に存在する仮想生命オブジェクトの移動範囲を規定するデータを記憶する記憶ステップと、仮想生命オブジェクトの共有仮想空間における新たな位置のデータを発生する発生ステップと、発生された位置を規定されている移動範囲内の位置に修正する修正ステップと、修正された位置に仮想生命オブジェクトを移動させる移動ステップとを備えるコンピュータプログラムを提供することを特徴とする。

【0015】請求項1に記載の共有仮想空間提供装置、請求項2に記載の共有仮想空間提供方法、および請求項3に記載の提供媒体においては、共有仮想空間内に存在する仮想生命オブジェクトの移動範囲を規定するデータが記憶され、仮想生命オブジェクトの共有仮想空間における新たな位置のデータが発生され、発生された位置が規定されている移動範囲内の位置に修正され、修正された位置に仮想生命オブジェクトが移動させる。

【0016】

【発明の実施の形態】以下に本発明の実施の形態を説明するが、特許請求の範囲に記載の発明の各手段と以下の実施の形態との対応関係を明らかにするために、各手段の後の括弧内に、対応する実施の形態（但し一例）を付加して本発明の特徴を記述すると、次のようになる。

【0017】請求項1に記載の共有仮想空間提供装置は、共有仮想空間内に存在する仮想生命オブジェクトの移動範囲を規定するデータを記憶する記憶手段（例えば、図27のステップ21）と、仮想生命オブジェクトの共有仮想空間における新たな位置のデータを発生する発生手段（例えば、図27のステップS23）と、発生された位置を規定されている移動範囲内の位置に修正する修正手段（例えば、図27のステップS26）と、修正された位置に仮想生命オブジェクトを移動させる移動手段（例えば、図27のステップS27）とを備えることを特徴とする。

【0018】以下、図面を参照し、本発明の実施の形態について説明する。

【0019】説明に先立ち、世界的規模で構築されたコンピュータネットワークであるインターネット(The Internet)において、様々な情報を提供するWWW(world wide web)の枠組みを利用して3次元的な情報を統一的に扱うことができる記述言語であるVRML(virtual reality modeling language)について説明する。

【0020】ここで、インターネットで利用できる情報提供システムとして、スイスのCERN(European Center for Nuclear Research:欧州核物理学研究所)が開発したWWWが知られている。これは、テキスト、画像、音声などの情報をハイパーテキスト形式で閲覧できるようにしたもので、HTTP(Hyper Text Transfer Protocol)と呼ばれるプロトコルに基づいて、WWWサーバに格納された情報をパーソナルコンピュータなどの端末に非同期転送するものである。

【0021】WWWサーバは、HTTPデーモン(HTTP:Hyper Text Transfer Protocol、ハイパーテキスト転送プロトコル)と呼ばれるサーバ・ソフトウェアとハイパーテキスト情報が格納されるHTMLファイルによって構成される。なお、デーモンとは、UNIX上で作業を行う場合、バックグラウンドで管理や処理を実行するプログラムを意味する。ハイパーテキスト情報は、HTML(HyperText Markup Language、ハイパーテキスト記述言語)と呼ばれる記述言語によって表現される。HTMLによるハイパーテキストの記述は、「<」と「>」で囲まれたタグと呼ばれる書式指定によって文章の論理的な構造が表現される。他の情報とのリンクについての記述は、アンカーと呼ばれるリンク情報によって行われる。アンカーによって情報が存在する所を指定する際には、URL(Uniform Resource Locator)が用いられる。

【0022】HTMLで記述されたファイルを、TCP/IP(Tra-

nsmission Control Protocol/Internet Protocol)ネットワーク上において転送するためのプロトコルがHTTPである。クライアントからの情報の要求をWWWサーバに伝え、HTMLファイルのハイパーテキスト情報をクライアントに転送する機能をもっている。

【0023】WWWを利用する環境として多く利用されているのが、WWWブラウザ(ブラウザとは閲覧するという意味)と呼ばれる、Netscape Navigator(米国Netscape Communications社の商標)をはじめとするクライアント・ソフトウェアである。

【0024】このWWWブラウザを用いて、URLに対応する世界規模で広がるインターネット上のWWWサーバ上のファイル、いわゆるホームページを閲覧することができ、いわゆるネットサーフィンと呼ばれるように、リンクが張られたホームページを次々と辿って、多種多様なWWWの情報ソースにアクセスすることができる。

【0025】近年、このWWWをさらに拡張し、3次元空間の記述や、3次元グラフィックスで描画されたオブジェクトに対してハイパーテキストのリンクの設定を可能とし、これらのリンクをたどりながらWWWサーバを次々とアクセスできるようにしたVRMLと呼ばれる3次元グラフィックス記述言語で記述された3次元空間を表示するVRMLブラウザが開発されている。

【0026】このVRMLの詳細は、例えば、「VRMLを知る:3次元電脳空間の構築とブラウジング[マーク・ペッシ著、松田晃一・蒲地輝尚・竹内彰一・本田康晃・暦本純一・石川真之・宮下健・原和弘訳、1996年3月25日初版発行、ブレンティスホール出版ISBN4-931356-37-0](原著:VRML:Browsing & Building Cyberspace,Mark Pesce,1995 New Readers Publishing ISBN 1-56205-498-8)」、および「VRMLの最新動向とCyber Passage[松田晃一・本田康晃著、bit(共立出版)/1996 Vol.28 No.7 pp29 乃至pp36, No.8 pp57 乃至pp65, No.9pp29 乃至pp36, No.10 pp49乃至pp58]」等の文献に記載されている。

【0027】また、August 4, 1996におけるThe Virtual Reality Modeling Language Version 2.0, ISO/IEC CD 14772の公式かつ完全な仕様書は、<http://webspac.e.sgi.com/moving-worlds/spec/index.html>で公開されており、その日本語版は、<http://www.webcity.co.jp/info/andoh/VRML/vrml2.0/spec-jp/index.html>で公開されている。

【0028】さらに、VRML 2.0用ブラウザおよび共有サーバ用ソフトウェアとしては、例えば、本出願人であるソニー株式会社が「Community Place Browser / Bureau(商標)」として開発し、製品化しており、そのβ版(試供版)を、インターネット上のホームページ<http://vs.sony.co.jp>からダウンロード可能としている。

【0029】このようなVRML 2.0を用いて3次元的な仮想空間を構築しようとする場合、まず、VRMLにより仮

想空間内の物体（モデル）の形、動きおよび位置等を示す図形データの作成（モデル作成）、ユーザが画面表示された仮想空間内のモデルを、例えば、マウスでクリックしてポインティングした場合にイベントを発生させるスイッチ（センサ）のモデルへの付加（センサ付加）、センサへのポインティングに応じて発生するイベントを実現するスクリプトのプログラミング（スクリプト作成）、センサーに対する操作とスクリプトの起動等、図形データおよびスクリプト（以下、図形データ、スクリプトおよびVRMLに規定されているライト等のコモンノード等を総称してノードとも記す）の間の対応付け（ルーティング）などによって所望のコンテンツを表現するVRMLファイルを作成する。

【0030】例えば、<http://www.ses.co.jp/SES/STAFF/kan/howto/howto1.html>には、VRML2.0の書き方、サンプルデータなど、丁寧に解説されている。その一部を紹介すると、以下の通りである。

【0031】1. VRML2.0で書かれたワールドを見るにはHTML形式で書かれたデータを見るためにHTMLブラウザが必要のように、VRML2.0形式で書かれたデータを見るためにはVRML2.0に対応したVRMLブラウザが必要となる。尚、このページで作成したVRML2.0のデータは全てSONY社製のCommunity Place Browserで行なわれている。また、ファイルがVRMLで書かれたことを表すためにファイルの拡張子を“.wrl”（ワールドの意）に、更にVRML2.0で書かれたことを表すためにファイルの1行目に
#VRML V2.0 utf8
と書くことが必要である。

【0032】2. VRML2.0データの基本構造
VRML2.0のデータはノード(Node)とフィールド(Field)で構成されていて、基本的に次のような形で書かれている。

Node { Field(s) }

この中でFieldsは省略することができるが、Nodeと中かっこ'{'、'}'は省略することができない。フィールドはノードに変数を渡し、ノードのパラメータを指定しま

Sample2

```
1: #VRML V2.0 utf8
2: Transform {
3:   children [
4:     Shape {
5:       appearance Appearance {
6:         material Material { diffuseColor 1 0 0 }
7:       }
8:       geometry Sphere { radius 1 }
9:     }
10:   ]
11: }
```

【0037】この球は赤色で表示される。追加されたのは5行目乃至7行目までである。diffuseColorフィールド

す。フィールドを省略した場合、デフォルト値が用いられる。また、フィールドには単一の値しか持たない「単一値フィールド(SF)」と複数の値を持つ「複値フィールド(MF)」があり、単一値フィールドの名前は“SF”で、複値フィールドは“MF”で始まるようになっている。

【0033】3. 球の書き方

VRML2.0では球や直方体、円柱、円錐などの基本図形を描くためのノードが用意されている。前にも書いているが、それぞれのノードはパラメータを指定するためのフィールドを持っている。例えば、球を描くためのSphereノードは半径を指定するためのradiusというフィールドを持っており、次のように書くことで半径1の球を表示する。尚、radiusフィールドのデータタイプは、SFFloatですから、1つの浮動小数点数の値をとる。

【0034】Sample1

```
1: #VRML V2.0 utf8
2: Transform {
3:   children [
4:     Shape {
5:       geometry Sphere { radius 1 }
6:     }
7:   ]
8: }
```

【0035】実は2,3行目、7,8行目は書かなくてもかまいません。1行目と4行目乃至6行目だけで、きちんと半径1の球を表示できる。Transformノードは、Groupノードという種類のノードの1つで、文字通りノードのグループ化を行うためのノードである。Transformノードを含めて、後でくる他のノードの詳しい機能やフィールドなどは「Appendix1: VRML2.0 Node List」を参照されたい。球を書くためのノードはSphereであるが、このノードはGeometryノードと呼ばれるノードの1つである。Geometryノードは、見え方と形状を定義するShapeノードのgeometryフィールドに書く必要がある。

【0036】4. 球に色を付ける

先ほどの球に色を付けるには次のように書く。

のデータタイプは、SFCOLORであるから、RGBカラーを表す3つの単精度浮動小数点数の組を1つだけ持つ。見え

方を定義するMaterialノードはAppearanceノードのmaterialフィールドに書くことになっています。また、AppearanceノードはShapeノードのappearanceフィールドに書くことになっている。このためこのような一見複雑な構造になっている。

Sample3

```
1: #VRML V2.0 utf8
2: Transform {
3:   children [
4:     Shape {
5:       appearance Appearance {
6:         texture ImageTexture { url "image.gif" }
7:       }
8:       geometry Box {}
9:     }
10:   ]
11: }
```

【0039】テクスチャを張り付けているのは6行目である。ImageTextureノードはAppearanceノードのtextureフィールドに書くことになっているため、このような書き方になる。尚、8行目のBoxノードは直方体を描くため

【0038】5. テクスチャの張り付け

オブジェクトに色を付けるだけではなくて画像ファイルを張り付けることもできる。VRML2.0でテクスチャとして利用できるファイルフォーマットはJPEG, GIF, PNGである。ここでは予め用意したGIFイメージを使っている。

のノードである。

【0040】6. 物体の位置を移動させる 次にこの赤色の球を右に移動させる。

Sample4

```
1: #VRML V2.0 utf8
2: Transform {
3:   translation 2 0 0
4:   children [
5:     Shape {
6:       appearance Appearance {
7:         material Material { diffuseColor 1 0 0 }
8:       }
9:       geometry Sphere { radius 1 }
10:     }
11:   ]
12: }
```

【0041】3行目に追加したtranslationフィールドで物体を平行移動させる。translationフィールドは、translation x y zで、それぞれx軸、y軸、z軸の移動量を指定する。ブラウザ上ではx軸は左右（右方向に+）、y軸は上下（上方向に+）、z軸は奥行き（手前に向かって+）を表す。従

って、translation 2 0 0は右方向に2だけ平行移動することになる。

【0042】7. 物体の追加 赤い球の左に緑の円錐を追加する。

Sample5

```
1: #VRML V2.0 utf8
2: Transform {
3:   translation 2 0 0
4:   children [
5:     Shape {
6:       appearance Appearance {
7:         material Material { diffuseColor 1 0 0 }
8:       }
9:       geometry Sphere { radius 1 }
```

```

10:     }
11:   ]
12: }
13: Transform {
14:   translation -2 0 0
15:   children [
16:     Shape {
17:       appearance Appearance {
18:         material Material { diffuseColor 0 1 0 }
19:       }
20:       geometry Cylinder {}
21:     }
22:   ]
23: }

```

【0043】13行目以降に追加したソースは12行目以前のものと同様である。物体が球から円錐へ、色が赤から緑へ、そして位置が左に移動している点が12行目以前のものとは異なる。

【0044】8. ポリゴンで書く
 上述した「VRML2.0の書き方」ではプリミティブな図形を使ったが、複雑な図形を書くときにはポリゴンを使用する。ポリゴンで表示するためにIndexedLineSet、IndexedFaceSetの2つのノードが用意されている。IndexedLineSetは線を、IndexedFaceSetは面を表現する。

Sample6

```

1: #VRML V2.0 utf8
2:   Transform {
3:     children [
4:       Shape {
5:         geometry IndexedFaceSet {
6:           coord Coordinate {
7:             point [
8:               0 0 0,
9:               1 0 0,
10:              1 0 1,
11:              0 0 1,
12:              0 1 0,
13:              1 1 0,
14:              1 1 1,
15:              0 1 1
16:            ]
17:          }

```

Sample7

```

1: #VRML V2.0 utf8
2: Transform {
3:   children [
4:     Shape {
5:       appearance Appearance {
6:         material Material { diffuseColor 0 0 1 }
7:       }

```

```

18:       coordIndex [
19:         0, 1, 2, 3, -1,
20:         0, 1, 5, 4, -1,
21:         1, 2, 6, 5, -1,
22:         2, 3, 7, 6, -1,
23:         3, 0, 4, 7, -1,
24:         4, 5, 6, 7, -1
25:       ]
26:       solid FALSE
27:     }
28:   ]
29: }
30: ]
31: }

```

【0045】このサンプルは6つの面で立方体を表している。ポリゴンで表すためにはまず、頂点となる座標を決める（7行目乃至16行目）。この座標は上から0,1,2,と番号が振られる。即ち、10行目の「1 0 1」は「2」番の座標ということになる。次に何番の座標と何番の座標で面を構成するのかが決める（18行目乃至25行目）。19行目の「0, 1, 2, 3, -1」は「0番、1番、2番、3番の座標で面を作る」ことを表している。

【0046】9. ノードに名前を付ける
 既に定義されているノードを再利用するためにDEFとUSEという機能がある。例えば、半径2の青の球を2つ描くとする。これまでの書き方で書くならば次のようになる。

```

8:         geometry Sphere { radius 2 }
9:     }
10: ]
11: }
12: Transform {
13:     translation 0 5 0
14:     children [
15:         Shape {
16:             appearance Appearance {
17:                 material Material { diffuseColor 0 0 1 }
18:             }
19:             geometry Sphere { radius 2 }
20:         }
21:     ]
22: }

```

【0047】12行目以降は13行目の座標移動の部分を除いて11行目以前と全く同じである。1度定義されている「半径2で青い球」を再び同じように記述するのははっ

きり言って無駄である。そこで次のようにする。

【0048】

Sample7改

```

1: #VRML V2.0 utf8
2: Transform {
3:     children [
4:         DEF BlueSphere Shape {
5:             appearance Appearance {
6:                 material Material { diffuseColor 0 0 1 }
7:             }
8:             geometry Sphere { radius 2 }
9:         }
10:     ]
11: }
12: Transform {
13:     translation 0 5 0
14:     children [
15:         USE BlueSphere
16:     ]
17: ]
18: }

```

4行目を見ると

DEF BlueSphere Shape

となっている。これは「Shape{……}を「BlueSphere」という名前にします。」という意味で、これ以降

USE BlueSphere

と書くだけでShape{……}と同じ内容を表す。

【0049】10. WRLファイルの読み込み

大きなVRMLデータをつくるときに1つのファイルにすべての記述を行うのはあまりいい方法ではない。部品ごとに分けて必要に応じて呼び出せるようにしておく方が便利である。このような場合、Inlineノードを使う。例えば1. でつくったSample1.wrlを読み込んで表示する。

Sample8

```

1: #VRML V2.0 utf8
2: Inline {
3:     url [
4:         sample1.wrl
5:     ]
6: }

```

【0050】11. リンクを張る

オブジェクトにリンクを張り、オブジェクトをクリックしたときに別のページにジャンプさせることもできる。

Sample9

```

1: #VRML V2.0 utf8
2: Anchor {
3:     children [

```



```

4:      Shape {
5:          geometry Sphere { radius 2 }
6:      }
7:  ]
8:  url "test.html"
9:  description "LINK to test.html"
10: }

```

【0051】リンクを張るオブジェクトを子ノードにしてAnchorノードで括る。AnchorノードはGroupノードの1つである。リンク先のファイルをurlフィールドに書く。descriptionフィールドに書いた文字列はマウスポインタがオブジェクトに触れている間表示される。

【0052】12. ライトの設定

VRML2.0ではワールドに光を設定するためにDirectional Light (平行光)、PointLight (点光源)、SpotLight (スポットライト) 3つのノードが用意されている。ここではPointLightノードを例に挙げて説明する。光の当たり方がわかりやすいように3行目でオブジェクトを回転させている。

Sample10

```

1: #VRML V2.0 utf8
2: Transform {

```

```

      Sample11
1: #VRML V2.0 utf8
2: NavigationInfo {
3:   headlight FALSE
4: }
5: Transform{
6:   children [
7:     Shape {
8:       appearance Appearance {
9:         material Material { diffuseColor 1 0 0 }
10:      }
11:     geometry Sphere {}
12:   }
13: ]
14: }

```

【0055】このサンプルでブラウザの Option を見ると、Headlight のチェックがはずれている。このサンプルでは今までのサンプルに比べて極端に暗くなっているのが分かる。HeadLight は常にユーザの見ている方向に照らされるライトで、これがないとこのように見える。新たに加えたのは 2 行目から 4 行目の NavigationInfo ノードである。このノードの headlight フィールドを TRUE か FALSE にすることで HeadLight を ON・OFF する。HeadLight を消して任意のライトを設定することで効果的に明るさを設定できる。

Sample12

```

1: #VRML V2.0 utf8
2: NavigationInfo {

```

```

3: rotation 1 0 0 0.75
4:   children [
5:     PointLight {
6:       location 0 0 2 #オブジェクトの前
7:     }
8:     Shape {
9:       geometry Cylinder {}
10:    }
11:  ]
12: }

```

【0053】6行目の光源の位置によってオブジェクトの見え方が異なる。このサンプルではオブジェクトの前に光源を置いている。

【0054】13. ワールド環境(1)

これまでは主にオブジェクトの作成に関しての解説であったが、今回はオブジェクト作成以外のノードの使い方について触れる。ワールド環境などと勝手に命名しているが、こういう言い方が一般的かどうかは分からない。まず、ブラウザでデフォルトで用意されている HeadLight を消してみる。HeadLight の ON・OFF はブラウザの Option で変更できるが、ファイル内に書くことによっても設定できる。

【0056】14. ワールド環境(2)

NavigationInfo ノードには他にも幾つかのフィールドが用意されている。その中の type フィールドでナビゲーション方法を変更することができる。デフォルトでは WALK に設定されている ナビゲーション方法であるが、他にも重力を無視して移動できる FLY、自分が動くことなくオブジェクトを動かす EXAMINE、何もコントロールできない NONE、がある。ちなみに WALKは重力の影響を受けながら移動するナビゲーション方法である。

【0057】

```

3:   type EXAMINE
4: }
5: Transform{
6:   children [
7:     Shape {
8:       appearance Appearance {
9:         material Material { diffuseColor 1 0 0 }
10:      }
11:     geometry Box {}
12:   }
13: ]
14: }

```

このサンプルでは EXAMINE にしている。オブジェクトをマウスでドラッグするとオブジェクトが回転する。

【0058】15. シーンにタイトルを付ける
HTML では <title> タグで囲むことによってタイトルを付けることができる。もしこのタグを指定しないとタイトルが [http://ryo1.is.kochi-u...howto3.html] のよ

うにパスが表示される。VRML でもこれと同じことが起こる。これまでのサンプルではタイトルを指定していないためパスが表示されている。VRML でタイトルを指定するには WorldInfo ノードを使用する。

【0059】

```

Sample13
1: #VRML V2.0 utf8
2: WorldInfo {
3:   title "Spin Box"
4:   info["Autor H. Kan", "http://ryo1.is.kochi-u.ac.jp/"]
5: }
2: NavigationInfo {
3:   type EXAMINE
4: }
5: Transform{
6:   children [
7:     Shape {
8:       appearance Appearance {
9:         material Material { diffuseColor 1 0 0 }
10:      }
11:     geometry Box {}
12:   }
13: ]
14: }

```

【0060】2 行目乃至 5 行目までに WorldInfo ノードを追加した。このサンプルでは前のサンプルに "Spin Box" というタイトルを付けた (Plugin 版では表示されなかった)。尚、info フィールドにはタイトル以外の情報を書くが、ブラウザ上には何の変化も与えない。

【0061】16. 視点の位置を変える

デフォルトでは視点は最初 z 軸上のどこか (オブジェクトの配置によって異なる) にいる。ここでは最初の視点位置を任意の場所に変更できるようにする。

【0062】

```

Sample14
1: #VRML V2.0 utf8
2: Viewpoint{ position x y z }
3: Transform {
4:   children [
5:     Shape {
6:       appearance Appearance {
7:         material Material { diffuseColor 1 0 0 }

```

```

8:      }
9:      geometry Sphere {}
10:    }
11:  ]
12: }
13: Transform {
14:   translation -3 0 0
15:   children [
16:     Shape {
17:       appearance Appearance {
18:         material Material { diffuseColor 0 1 0 }
19:       }
20:       geometry Sphere {}
21:     }
22:   ]
23: }
24: Transform {
25:   translation 3 0 0
26:   children [
27:     Shape {
28:       appearance Appearance {
29:         material Material { diffuseColor 0 0 1 }
30:       }
31:       geometry Sphere {}
32:     }
33:   ]
34: }

```

【0063】(0, 0, 0)に赤、(-3, 0, 0)に緑、(3, 0, 0)に青の球がある。2行目の Viewpoint ノードの position フィールドに具体的な座標を指定すると、そこが最初の視点になる。ただし視線方向は常に z 軸の方向である。

【0064】17. 視線の方向を変える

sample14 では視点の座標のみを変更したが、もちろん視線方向も指定できる。方向を指定するときも Viewpoint ノードを使うが、フィールドは orientation フィールドを使う。orientation フィールドは SFRotation 型のフィールドで、加算軸と回転角を値として持つ。

【0065】18. センサをつける

VRML2.0 のノードの中には Sensor ノードと呼ばれるノードがある。シーン内での様々なことを感知、判定するためのノードで、全部で7種類用意されている。このWeb ページで動作確認を行っている Community Place Browser では、幾つかの Sensor ノードがまだサポートされていない。ここではオブジェクトにマウスがヒットしたかどうかを感知する TouchSensor をオブジェクトにつけてみる。

【0066】Sample15

```

1: #VRML V2.0 utf8
2: Transform{

```

```

3:   children [
4:     DEF TS TouchSensor {}
5:     Shape {
6:       geometry Box {}
7:     }
8:   ]
9: }

```

【0067】これまでのサンプルと異なるのは4行目だけである。ここでは TS という名前を付けた TouchSensor を Box につけている。Box にマウスカーソルが触れると(あなたが使っているブラウザが Community Place Browser ならば)手の形に変わるはずである(他のブラウザの中にはカーソルが変わらないものもある)。ちなみにクリックしても何も起きない。

【0068】19. 動きをつける(1)

VRML2.0 が VRML1.0 と大きく異なる点はシーンに動きをつけられるという点である。動きをつけるには Java や VRMLScript(JavaScript)などのスクリプトを用いる方法と、Interpolator ノードを用いる方法がある。まずは Interpolator ノードを用いる方法から見ていく。Interpolator とは、「差し挟む」とか「内挿する」といった意味である。Interpolator ノードの数値、位置、3D 座標、方向、法線、色の値を変えることによってシーン

に動きをつけることができる。ここでは方向を内挿する
OrientationInterpolator ノードを用いてオブジェク

トを回転させてみる。
【0069】

```

Sample16
1: #VRML V2.0 utf8
2: DEF OBJ Transform{
3:   children [
4:     Shape {
5:       geometry Box { size 2 3 1 }
6:     }
7:   ]
8: }
9: DEF TS TimeSensor{
10:   cycleInterval 1
11:   loop TRUE
12:   stopTime -1
13: }
14: DEF OI OrientationInterpolator{
15:   key [0, 0.125, 0.25, 0.375, 0.5,
16:       0.625, 0.75, 0.875, 1,]
17:   keyValue [0 1 0 0, 0 1 0 0.785, 0 1 0 1.57,
18:            0 1 0 2.355, 0 1 0 3.14, 0 1 0 -2.355,
19:            0 1 0 -1.57, 0 1 0 -0.785, 0 1 0 0]
20: }
21: ROUTE TS.fraction_changed TO OI.set_fraction
22: ROUTE OI.value_changed TO OBJ.set_rotation

```

【0070】まず、幾つかのノードに名前を付けておく。2, 9, 14 行目にそれぞれ、OBJ, TS, OI と定義している。これは後述するがイベントの受け渡しをするときに必要になる。9 行目乃至 13 行目を見ます。TouchSensor は時間の経過を感知するノードで、時間の経過に伴って一定の間隔でイベントを生成することができる。loop フィールドは TRUE か FALSE をとる SFBool フィールドで、TRUE なら stopTime になるまで続く。ここでは stopTime -1 と startTime (デフォルトは0) と startTime より小さな値になっているので永久に継続されることになる。回転する動きをもっと遅くするときには cycleInterval の値をもっと大きくする。

【0071】14 行目乃至 20 行目までが OrientationInterpolator ノードである。Interpolator ノードはすべて key と keyValue の2つのフィールドを持っている。keyはアニメーション時間の間隔を 0 から 1 の間で設定する。keyValue は keyで設定した間隔に具体的なフィールド値 (ここでは MFRotation) を設定する。ここではアニメーション時間の間隔を9等分して、y軸

を中心に回転角を設定している。

【0072】しかしこれだけではシーンに動きをつけることはできない。TimeSensor ノードの生成したイベントを OrientationInterpolator ノードに渡してやる必要がある。21, 22行目を見ます。この ROUTE というキーワードで始まる行で イベントを受け渡しを行う。TimeSensor TS が起動すると fraction_changed がイベントアウトされる。fraction_changed がイベントアウトされると、OrientationInterpolator OI の set_fraction にイベントインされる。ここまでの21 行目の ROUTE の動作である。22 行目はイベントインされた set_fraction 値から OrientationInterpolator OI が値を内挿し、それを Transform OBJ の translationフィールドに value_changed としてイベントアウトする。もう1つサンプルをあげてみる。

【0073】20.動きをつける(2)

今度はマウスでオブジェクトをクリックすると移動するサンプルである。移動するには位置を内挿する PositionInterpolator ノードを使用する。

```

Sample17
1: #VRML V2.0 utf8
2: DEF OBJ Transform {
3:   children [
4:     DEF TS TouchSensor {}
5:     Shape {

```

```

6:         appearance Appearance {
7:             material Material { diffuseColor 1 0 0 }
8:         }
9:         geometry Box {}
10:     }
11: ]
12: }
13: DEF TIS TimeSensor { cycleInterval 5 }
14: DEF PI PositionInterpolator {
15:     key [0, 0.2, 0.4, 0.6, 0.8, 1.0]
16:     keyValue[0 0 0, 0 0 -2, 0 2 -4, 0 4 -6, 2 4 -6, 4 2 -6]
17: }
18: ROUTE TS.touchTime TO TIS.set_startTime
19: ROUTE TIS.fraction_changed TO PI.set_fraction
20: ROUTE PI.value_changed TO OBJ.set_translation

```

【0074】今度は、OBJ という名前の赤い立方体に TouchSensor TS をつけている。オブジェクトがクリックされると、TouchSensor TS の touchTime が TimeSensor TIS の startTime にイベントアウトされる。サンプルのソースには TimeSensor ノードの中に startTime フィールドは書いてないが、デフォルトで startTime 0 となっている。あとは前のサンプルと同じである。これらの ROUTE によってオブジェクトが移動する。

【0075】ここで、第1世代のVRML 1.0 に対し、第2世代のVRML 2.0 で新たに加わった機能であるVRML仮想空間内における自律的な動き(Behavior)を実現するためのメカニズムについて概説する。

【0076】VRML 2.0 においては、3次元仮想現実空間内に配置されたオブジェクトに対する操作に伴って生じるイベントや、予め設定された時間が経過した時点で生じるタイマーイベントに応じて、オブジェクトの自律的な動きを実現することができる。このBehaviorのメカニズムは、センサー、ルーティング、スクリプトの3つの要素の連携動作によって実現される。

【0077】すなわち、

①予め3次元仮想現実空間内に配置されたオブジェクトなどのノードに対して関係付けられて、VRMLファイルとして記述されたセンサーノードが、そのフィールドの値の変化に基づいて、外部イベントを感知し、VRMLシーン内にイベントを発生する。

②発生したイベントは、VRMLファイルとして記述されたルーティングに基づいて、オブジェクトの挙動を規定するプログラムである外部のスクリプトに対して、伝達される。

③外部のスクリプトには予め特定のイベントが到来した時点で呼び出されるメソッドが記述されており、ルーティングによって伝達されたイベントを受け取った外部のスクリプトは、その記述に基づく処理を実行した後、その処理結果と、ルーティングの記述に基づいて、VRMLシーン内の該当するノードのフィールドの値を変更する。

【0078】VRML 2.0 においては、センサーノードとしては、例えば、指定されたオブジェクト上をポインティングデバイスが通過したり、ユーザによってクリックされた場合にイベントを発生するTouchSensorや、指定された領域内にViewPoint(ユーザの視点)が侵入した場合にイベントを発生するProximitySensorや、与えられた時刻となったり、予め指定された時間間隔が経過する毎に発生するTimeSensorなどが定義されている。

【0079】もう少し詳しく、Behaviorの仕組みを説明する。先にも述べたように、Behaviorの仕組みは、センサー、イベント、ルーティングおよびスクリプトから構成されている。

【0080】センサーは、2つの機能に分けられる。

- ・ユーザの操作を感知するタイプ
- ・システムの変化を感知するタイプ

【0081】ユーザの操作を感知するタイプのセンサーは、3次元仮想空間内に配置された物体などに関係付けられたソフトウェアによるスイッチとなる。システムの変化を感知するタイプのセンサーは、作動させる時間を事前に設定しておいたタイマーが作動することになる。センサーの働きは、これらの外部イベントを検出して、それをVRML内部のイベントに変換することである。

【0082】イベントは、VRMLの内部の関連するノード間で、情報を伝えるためのデータのことを指す。実際には、VRMLファイルの中に記述されたフィールド値の変化がイベントとして伝達される。

【0083】ルーティングは、センサー機能で検出されたイベントをどのノードに伝えるかを指定するための仕組みで、イベントによる情報の伝達経路を指定することになる。

【0084】スクリプトはイベントの入出力口で、入力されたイベントから何らかの計算を行い、その結果をイベントとして出力することができるものである。スクリプトは特定の言語のみに制限されるものではなく、現段階では、インターネット界で注目を浴びているJavaやJa

vaScript、通常のシステムでよく利用されているC言語、UNIXでよく利用されているTcl/TkやPERL、マイクロソフト社の提供しているVisual Basic言語などが対応している。このようにVRML 2.0では特定のスクリプト言語に依存しない仕様になっている（VRML 2.0仕様検討中、一時特定の言語仕様としてVRMLScriptが採用されようとしたこともあるが、この構想は取り止めになった）。

【0085】次に、Behaviorの処理手段に関して、図1を参照して説明する。Behaviorの処理を図式化すると、図1のようになる。以下、各部に分けて、処理信号の流れを説明する。

【0086】①センサーノード

前にも述べたように、センサーノードには、大きく分類して2系統がある。ユーザの操作を感知するタイプと、システムの変化を感知するタイプのセンサーである。

【0087】前者のセンサーには、3次元物体やその平面をマウスクリックしたり、通過したりしたときに感知するTouchSensorやPlaneSensorなどのセンサーノードがあり、後者のセンサーとしては、設定した時間が来るとイベントを発生させる仕組みになっているTimeSensorが用意されている。

【0088】この図1の例では、球にTouchSensorが付けられているものとする。このとき、ユーザが球をマウスクリックすると、このイベントをTouchSensorが検出する。このイベントは、TouchSensorのeventOutフィールドのフィールド値が変化することで検出される。通常1回のマウスクリックで2つのイベントを発生する。それは、マウスボタンが押されたタイミングと、離されたタイミングである。

【0089】次に、このイベントは、ルーティング記述部分によって、ルーティングされることになる。

【0090】②ルーティング

このイベントのルーティングを指定するのが、図2に示すような「Route」である。

【0091】センサー記述部分で発生したイベントがRouteのeventOutフィールドに伝達され、さらに、次に述べるスクリプトノードに伝えられることにより、外部ファイルにイベントが渡され、Behavior機能が実行される。

【0092】③スクリプトノード

このノードは、VRMLファイルと外部のスクリプトとを連動させる仲介をするノードである。スクリプトノードの記述書式に従って、どのような言語で記述されているのかとか、ファイル名を指定したり、外部スクリプトファイルとの間でイベントのやり取りをするために、eventInフィールドとeventOutフィールドを規定する。このとき、使用できるスクリプトファイルは、Java、JavaScript、C言語、Tcl/Tk、PERL、およびVisual Basic言語など多岐に渡る。

【0093】実際の処理手段としては、ルーティングさ

れたイベントがスクリプトノードに記述されているスクリプトファイルに処理が伝達され、外部のスクリプトファイルが実行される。外部のスクリプトファイルは、その中で定義されているeventIn（イベントの入口）で受け取り、そのファイル内に記述された処理を実行する。その処理が終了した後、eventOut（イベントの出口）から結果をVRMLファイルのルーティングへ返す。VRMLファイルは、その返された結果を実行して、一連のBehavior処理は終了する。

【0094】このような、センサー、ルーティング、スクリプトの連携動作によって実現されるBehaviorのメカニズムを利用することによって、例えば、3次元仮想現実空間内に配置されたスイッチを模したオブジェクトをマウスでクリックすることによって、同じ空間内に配置された他のオブジェクトの外観（形状、姿勢、サイズ、色等）であるとか、挙動シーケンスなどを動的に変更する事ができる。

【0095】このBehaviorのメカニズムの詳細な説明については、<http://webpace.sgi.com/moving-worlds/spec/part1/concepts.html>および、その日本語版である<http://www.webcity.co.jp/info/andoh/VRML/vrml2.0/spec-jp/part1/concepts.html>で公開されている、August 4, 1996における The Virtual Reality Modeling Language Version 2.0, ISO/IEC CD 14772の仕様書、4. 概念の節に開示されている。この節には、VRML仕様書を利用するにあたりキーとなる概念が記述されている。ノードをシーングラフに結合する方法、ノードがイベントを生成したり受け取ったりする方法、プロトタイプによるノードタイプの作成方法、VRMLにノードタイプを追加して外部から使用できるようにエクスポートする方法、VRMLファイルにプログラムとして動作するスクリプトを組み込む方法など、様々なノードに関する一般的な項目が記載されている。

【0096】次に、このようなVRML 2.0の自律的な動き(Behavior)を実現するメカニズムを応用して、共有仮想空間内に、仮想生命オブジェクトを誕生させ、ユーザの操作や時間経過に伴う所定のイベントの発生に応じて変化する成長パラメータ（外観的成長又は内面的成長（個性））をサーバで管理し、このサーバから転送された成長パラメータに基づいて、仮想生命オブジェクトの外観（形状、姿勢、サイズ、色等）又は挙動シーケンスの内の何れか一方もしくは双方を動的に変更するためのスクリプトプログラムを解釈、実行することで、成長パラメータに応じた仮想生命オブジェクトを表示する技術について、以下に詳述する。

【0097】図3は本発明の一実施の形態の全体のシステム構成図である。

【0098】図3において、1, 2, 3は、VRMLブラウザ及びWWWブラウザがインストールされ、これらが動作しているクライアントPC（パーソナルコンピュータ）

であり、IP（インターネット接続サービスプロバイダ）4、5、6を介してインターネット7と接続されている。

【0099】インターネット7とルータ8を介して接続されたLAN(Local Area Network)9には、WWWサーバ10、WLS(World Location Server)11、共有サーバ12、AO(Application Object)サーバ13、14、メールサーバ15、およびコミュニケーションサーバ16が接続されている。これらの各サーバ10乃至16には、ハードディスク(HDD)10a、10b、11a乃至16aが、各々設けられている。

【0100】コミュニケーションサーバ16は、公衆電話回線網17を介して電話機18やファクシミリ19と接続され、さらに、PHS(Personal Handyphone System)サービスプロバイダ20を介してPHS端末23に無線接続され、ポケットベルサービスプロバイダ21を介してポケットベル端末24に無線接続されている。

【0101】図4はクライアントPC1のハードウェア構成を示すブロック図である。

【0102】図4において、30は各部を制御するCPU、31はVRML 2.0 fileや、Java(米国Sun Microsystems社の商標)による共有仮想生命の成長スクリプトプログラム等からなるVRMLコンテンツ、および飼い主データが格納されたHDD、32はCD-ROMディスク33に格納されたVRMLコンテンツを読み取るCD-ROMドライブ、34はBIOS(Basic Input Output Systems)等が格納されたROM、35はマイクロフォン36と左右のスピーカ37、38が接続されたサウンド処理回路、39はインターネット7に接続するためのMODEM、40はマウス41とキーボード42が接続されたI/O(入出力)インターフェイス、43はVRAM44が内蔵されたグラフィックス処理回路、45はCRTモニタ、46はRAMである。

【0103】このRAM46には、実行時に、Windows95(米国Microsoft社の商標)の上で動作するWWWブラウザであるNetscape Navigatorと、Javaインタプリタと、ソニー株式会社によって開発されたVRML 2.0ブラウザであるCommunity Place Browserが読み込まれて、CPU30によって実行される状態となっている。

【0104】VRML 2.0ブラウザには、米国シリコングラフィックス社によって開発され、無償公開されているVRMLの構文解釈用ライブラリ(パーサ)であるQvLibと、英国Criterion Software Ltd.のソフトウェア・レンダラであるRenderWare等、もしくはこれらと同等の機能を有するパーサやレンダラが実装されている。

【0105】そして、Community Place Browserは、図3に示すように、WWWブラウザとしてのNetscape Navigatorとの間において、NCAP(Netscape Client Application Programming Interface)(商標)に基づいて各種データの授受を行う。

【0106】Netscape Navigatorは、インターネット7

を介してWWWサーバ10よりHTMLファイルとVRMLコンテンツ(VRMLファイルとJavaによるスクリプトプログラムとを含む)の供給を受けると、これらをローカルのHDD31にそれぞれ記憶させる。Netscape Navigatorは、このうちのHTMLファイルを処理してテキストや画像をCRTモニタに表示する一方、Community Place BrowserはVRMLファイルを処理して3次元仮想空間をCRTモニタに表示するとともに、Javaインタプリタによるスクリプトプログラムの処理結果に応じて、3次元仮想空間内のオブジェクトの挙動を変化させる。

【0107】なお、図示は省略するが、他のクライアントPC2やクライアントPC3も、クライアントPC1と同様に構成されている。

【0108】次に上述した一実施の形態の動作について説明する。

【0109】まず、実際にVRMLコンテンツをインターネット経由でダウンロードしてから、1つの仮想空間を複数のユーザで共有するマルチユーザ環境とするまでの手順を図5乃至図7を参照して説明する。

【0110】図5において、番号1で示すように、最初に、WWWブラウザを用いて、VRMLコンテンツを提供しているWebサイトのホームページを閲覧する。この例では、<http://pc.sony.co.jp/sapari/>を閲覧している。次に、番号2で示すように、クライアントPC1とクライアントPC2のユーザは、VRML 2.0fileと、VRML空間内での自律的な動き(Behavior)を実現するためのスクリプトプログラム(Javaによる成長スクリプトプログラム)とからなるVRMLコンテンツを、それぞれダウンロードする。

【0111】勿論、CD-ROMディスク33で提供されるVRMLコンテンツをCD-ROMドライブ32で読み込んでも良い。

【0112】次に、図6に示すように、クライアントPC1及びクライアントPC2は、それぞれにダウンロードされ、一旦ローカルのHDD31に格納されたVRML 2.0fileを、VRML 2.0ブラウザであるCommunity Place Browserが解釈・実行し、さらに番号3で示すように、VSCP(Virtual Society Server Client Protocol)に基づいて、WLS11に対して共有サーバ12のURLを問い合わせる。このとき番号4で示すように、WLS11はHDD11aに格納された共有サーバURL管理テーブルを参照して、クライアントPC1及びクライアントPC2に対して、共有サーバ12のURLを通知する。

【0113】このURLを用いて、図7に示すように、クライアントPC1とクライアントPC2が、共有サーバ12に接続する。その結果、番号5で示すように、この共有サーバ12を介して共有3Dオブジェクトの位置や動きなどに関する共有メッセージの送信が行われ、番号6で示すように、その転送が行われ、マルチユーザ環境が実現される。

【0114】なお、以上の接続手順の詳しい説明については、特開平9-81781号公報を参照されたい。

【0115】次に、共有仮想空間内に存在する仮想生命オブジェクトの挙動を管理するAOサーバ13について説明する。AOサーバ13は、VSAPに基づいて共有サーバ12とAOに関するデータのやり取りを行うが、機能ごとに複数のAOに分類されている。例えば、ユニバーサルAOは、仮想空間に1つだけ存在し、仮想空間全体を覆うAOであり、全てのユーザのログイン、ログアウトを監視することができる。また、ユニバーサルAOは、Community Place Browserのアトリビュート機構を利用し、クライアントPCのレジストリの内容を把握することができたり、所定の情報をレジストリに書き込むことができたりする。さらに、ユニバーサルAOは、ユーザを特定するID（ユーザー人が一匹の仮想ペットを飼育できるときは、3DオブジェクトIDが相当する）は発行したり、また、そのIDを、例えば、成長パラメータ管理テーブルに登録したりすることができる。また、ユニバーサルAOは、共有仮想空間を利用しているユーザに同報通信（場内アナウンス）を行うことができる。

【0116】オブジェクトAOは、仮想空間内に落ちている（置かれている）仮想生命オブジェクト以外の物体である仮想物体オブジェクトを管理し、仮想物体オブジェクトがユーザによって拾われた場合、仮想物体オブジェクトは拾ったユーザの所有物として管理される。

【0117】エージェントAOは、ユーザが共有仮想空間で飼育することができる仮想ペットを管理するAOで、例えば、ユーザー人に対して、1つの仮想ペットを割り当てている。

【0118】エリアAOは、共有仮想空間の所定の領域に特殊な機能を設定するためのAOである。例えば、バトルエリアにおいては、仮想生命オブジェクト同士を戦わせることができる。また、チャットエリアにおいては、チャットログをすべて保存し、過去の発言のテキストを自由に取り出して、現在の会話に利用することができる。また、トラッシュ・リサイクルエリアにおいては、このエリアにオブジェクトが置かれると、オブジェクトが捨てられたことになる。

【0119】なお、オブジェクトAO、エージェントAO、およびエリアAOを含むすべてのAOは、ユニバーサルAOにより生成され、管理されている。

【0120】図8に示すように、成長パラメータ管理テーブルの仮想生命オブジェクトに関するデータは、仮想生命データと、飼い主データに大別される。

【0121】仮想生命データは、1つの共有仮想空間内において3Dオブジェクトを一意に特定するための3DオブジェクトID、共有仮想空間内での仮想生命オブジェクトの3次元座標値、飼い主によって選ばれた種や猫などの生物の種類、その性別、飼い主によって付与されたニックネーム、飼い主によって初期設定された日時、すなわ

ちその仮想生命オブジェクトの誕生日時、誕生した仮想空間に付与されているワールド名（本籍ワールド名）と、仮想生命オブジェクトの成長パラメータとからなる。

【0122】成長パラメータは、仮想生命の外観的成長を規定するフィジカルパラメータと、性格等が反映された内面的成長を規定するためのメンタルパラメータに大別される。

【0123】フィジカルパラメータは、身長（単位cm）、体重（単位Kg）、体格指数、食欲指数、健康度指数、および寿命残時間（単位時間）とからなる。

【0124】メンタルパラメータは、知能指数、言語能力指数、社交性指数、自主性指数、活発性指数、および機嫌指数により構成されている。

【0125】これらのパラメータは、飼い主によって最初に初期設定された誕生日時からの経過時間に伴って発生するタイマーイベントと、クライアントPCからのコールメッセージや操作メッセージに伴って発生するアクセスイベントや操作イベントに基づき、所定の成長パラメータ算出式によって算出された値に順次更新される。

【0126】図9は、クライアントPC1のCRTモニタ45の画面上のVRMLブラウザのメインウインドウに隣接して表示されるアクションパネルの機能を示している。

【0127】この図において、(Active)と表記されているAは「呼ぶボタン」であり、仮想ペットを呼んだり、寝ている仮想ペットを起こす際にクリックされる。

【0128】(Sleep)と表記されているBは「寝かすボタン」であり、仮想ペットを寝かす際にクリックされる。

【0129】Cは「食事ボタン」であり、仮想ペットに食事を与える際にクリックされるボタンである。

【0130】Dは「ほめるボタン」であり、仮想ペットに笑いかけて褒めてあげる際にクリックされるボタンである。

【0131】Eは「あそぶボタン」であり、飼い主が鬼になって逃げ回る仮想ペットを追いかけて、壁にぶつかって逃げられない仮想ペットにぶつかるまで追いつめて遊ぶ、いわゆる鬼ごっこで遊ぶ際にクリックされるボタンである。

【0132】Fは「しかるボタン」であり、言う事を聞かない仮想ペットを叱り付けて、しつけを行なう際にクリックされるボタンである。

【0133】Gは「きれいにするボタン」であり、仮想ペットをブラッシングしてきれいにする際にクリックされるボタンである。

【0134】そして、例えば、図10に示すように、自分のクライアントPC1において、「呼ぶボタン」Aがクリックされ（アクションパネルが操作され）、そのコールメッセージが共有サーバ12を介してAOサーバ13へ送信されると（ステップS1）、そのアクセスイベントに基づいて成長パラメータ管理テーブルの成長パラ

メータ更新処理が実行される（ステップS3）。このアクセスイベントに基づいて、食欲指数、健康度指数、機嫌指数が、1/10から10/10まで、0.1ポイントづつインクリメントされる。

【0135】また、例えば、「食事ボタン」（がクリックされ、その操作メッセージがAOサーバ13へ送信され（ステップS2）、操作イベントが発生する毎に、成長パラメータの体重が増加し、それに伴って、体格指数が、1/10から10/10まで、0.1ポイントづつインクリメントされる（ステップS3）。

【0136】その後、時間経過に伴うタイマーイベントが発生すると、成長パラメータの体重が減少し、それに伴って、体格指数が0.1ポイントづつデクリメントされる（ステップS3）。

【0137】例えば、この体格指数を含む成長パラメータは、更新される毎に、共有サーバ12のマルチキャスト処理（ステップS5）により、元の飼い主のクライアントPC1と、仮想空間を共有しているその他のクライアントPC2へ転送される（ステップS4）。

【0138】クライアントPC1では、返送されてきた成長パラメータに基づいて、仮想ペットの成長に伴う自律的な挙動を制御するための処理手順が記述された成長スクリプトプログラムが実行され（ステップS6）、VRMLファイルの仮想ペットを表現するための3Dオブジェクトを構成する各ノードのフィールドの値が変更され（ステップS7）、この変更されたフィールドの値が反映された仮想ペットがレンダリングされ（ステップS8）、クライアントPC1のCRTモニタ45の画面上のVRMLブラウザのメインウィンドウ上に表示される。

【0139】このクライアントPC1と同じ処理が、仮想空間を共有しているその他のクライアントPC2においても実行され、これにより仮想ペットの成長に伴って変更されたフィールドの値が反映された仮想ペットがレンダリングされ、他のクライアントPC2のCRTモニタ画面上のVRMLブラウザのメインウィンドウ上にも表示されることになる。

【0140】図11及び図12に、VRMLファイルの仮想ペットを表現するための3Dオブジェクトを構成する各ノードに対応したpart0乃至part5の関係（図11）と、その表示例（図12）を示す。part0が仮想ペットの頭部に相当し、part1が仮想ペットの胴体に相当し、part2とpart3が仮想ペットの右腕と左腕に相当し、part4とpart5が仮想ペットの右足と左足に相当する。

【0141】これらの各part0乃至part5に対応する各ノードのフィールドの値を変更することで、仮想ペットの各部の外観（形状、姿勢（向き）、サイズ、色等）、及び各部の挙動シーケンスを動的に変更する事が出来る。これは全て、成長パラメータに基づく成長スクリプトプログラムの処理で実現される。すなわち、VRML2.0で定義されたセンサー、ルーティング、スクリプトの連携動

作によって実現されるBehaviorのメカニズムを利用することによって実現される。

【0142】従って、従来の携帯電子ペットの仮想生物の画像表示方法のように、予め仮想生物のキャラクタの成長過程の各段階のビットマップ画像をROMに格納しておく必要はなく、例えば、仮想ペットの成長経過に応じてその体格や挙動を連続的かつ動的に変化させる事が出来る。

【0143】図13は、仮想ペットの成長と体格指数の遷移に伴って仮想ペットの体格を動的に変更して表示する概念図を示している。加齢に伴って、顔が大人の顔となり、体格も大きくなるが、体格指数が小さいと、やせた身体となり、大きいと、大きな身体となる。

【0144】図14は、仮想ペットの機嫌指数の遷移に伴って仮想ペットの表情を動的に変更して表示する概念図を示している。機嫌指数が大きいと、笑顔となり、小さいと、怒った顔になる。

【0145】図15は、仮想ペットの活発性指数の遷移に伴って仮想ペットの各部の挙動シーケンスを動的に変更して表示する概念図を示している。活発性指数が小さいと、足の屈伸程度の動きしかできないが、大きくなると、手を振ったり、頭を振ったりすることができる。

【0146】図16は、仮想ペットの知能指数の遷移に伴って仮想ペットへ頭髪を付加したり眼鏡を付加して表示する概念図を示している。

【0147】成長パラメータの知能指数は、図9に示す「呼ぶボタン」Aの操作に伴うアクセスイベントに基づいて、0.1ポイントづつインクリメントされ、図16に示すように仮想ペットの外観に影響を与える。

【0148】言語指数は、図9に示す「呼ぶボタン」Aの操作に伴うアクセスイベントやタイマーイベントに基づく仮想ペットの年齢に応じて0.1ポイントづつインクリメントされ、仮想ペットの年齢増加に応じて、テキストベースでのチャットの文章自動生成処理を実行するに際し、そのテキストの文体に影響を与える。例えばポイントの小さい仮想ペットのチャットは、ひらがなまたはカタカナで行われ、ポイントの大きな仮想ペットのチャットは、漢字の入った文字で行われる。

【0149】社交性指数は、飼い主とのチャットの頻度に応じて0.1ポイントづつインクリメント又はデクリメントされ、頻度が多ければ外向的な振る舞い、頻度が少なくなれば内向的な振る舞い、となるように仮想ペットの振る舞いに影響を与える。社交的で明るく、積極的な性格の仮想ペットは、姿勢も顔色も良くなり、逆に、内向的で暗く、消極的な性格の仮想ペットは、姿勢も顔色も悪くなる。

【0150】自主性指数は、タイマーイベントに基づく仮想ペットの年齢増加に応じて、0.1ポイントづつインクリメントされ、次第に飼い主の言う事を聞かなくなるなどの仮想ペットの振る舞いに影響を与える。

【0151】活発性指数は、年齢、食欲指数、健康度指数などに基づいて決定され、図15に示すように、仮想ペットの振る舞いに影響を与える。また、図9に示す「あそびボタン」Eの操作に伴う操作イベントの発生に基づいて、活発性指数が0.1ポイントずつインクリメントされ、次第に逃げ足が早くなるなどの仮想ペットの振る舞いに影響を与えたり、仮想ペットの体重を減らして体格指数をデクリメントして図13に示すようにその外観に動的な変更を与える。

【0152】機嫌指数は、図9に示す「呼ぶボタン」Aの操作に伴うアクセスイベントやタイマーイベントに基づくアクセス頻度などに基づいて決定され、図14に示すように、仮想ペットの表情に影響を与える。

【0153】一方、図8に示す成長パラメータ管理テーブルの飼い主データは、飼い主の氏名、飼い主への連絡手段（連絡方法）、および、その連絡先からなる。

【0154】連絡手段が0の場合、飼い主への連絡は、インターネット7経由の電子メールによるメッセージ文によって行われる。連絡手段が1の場合は通常のアナログ音声の電話機18に対して、コミュニケーションサーバ16でメッセージ文のテキストデータを自動読み上げツールで音声に変換して通知する。連絡手段が2の場合は、PIAFS(PHSインターネットアクセスフォーラム標準)のデータ伝送方式に準拠した電子メールサービスを利用し、メッセージ文によってPHS端末23へ通知する。連絡手段が3の場合は、ファクシミリ19に対して文書で通知する。連絡手段が4の場合は、ポケットベル端末24に対してメッセージ文で通知する。

【0155】このような飼い主データを管理するのは、後述する既存の通信インフラストラクチャを利用した飼い主への通知機能や、既存の通信インフラストラクチャを利用した仮想ペットへの簡易的な操作機能を実現する為である。

【0156】以上のシステムをまとめると、次のようになる。すなわち、共有仮想空間内に存在する仮想生命オブジェクトの、所定イベント（ユーザの操作又は時間経過に伴うイベント）の発生に応じて変化する成長パラメータ（外観的成長または内面的成長（個性））をAOサーバ13で管理し、このAOサーバ13から転送された成長パラメータに基づいて、クライアントPC1、2で、仮想生命オブジェクトの外観（形状、姿勢、サイズまたは色）又は挙動シーケンスの内の何れか一方もしくは双方を動的に変更するスクリプト（プログラム）を解釈し、成長パラメータに応じた仮想生命オブジェクトを表示する。

【0157】共有仮想空間内に存在する仮想生命オブジェクトの自律的な挙動を管理するAOサーバ13に、仮想生命オブジェクトの、所定イベント（ユーザの操作又は時間経過に伴うイベント）の発生に応じて変化する成長パラメータを管理する管理テーブルを設け、クライア

ントからの要求または所定イベントの発生に応じて管理テーブルから読み出した成長パラメータを、要求元またはその他のクライアントの内の何れか一方もしくは双方のクライアントへ返送する。

【0158】成長パラメータは、上述したように、仮想生命オブジェクトの誕生からの所定イベント（ユーザの操作又は時間経過に伴うイベント）の発生に基づいて算出される、外観的な成長度合いを示す値である。従って、例えば、よちよち歩きの赤ん坊から立派な大人を経て老人になるまでの年齢に応じた仮想生物や仮想ペット（AO）の外観上の変化が外観的成長パラメータで規定される。

【0159】成長パラメータは、仮想生命オブジェクトに対する所定イベント（ユーザの操作又は時間経過に伴うイベント）の発生に起因して算出される、内面的成長度合い（性格）を示す値でもある。例えば、社会的で明るく積極的な性格の仮想生命オブジェクトは姿勢も顔色も良くなり、逆に内向的で暗く消極的な性格の仮想生命オブジェクトは姿勢も顔色も悪くなるなどの性格に応じた仮想生物や仮想ペット（AO）の変化が内面的成長パラメータで規定される。

【0160】内面的成長パラメータは、仮想生命オブジェクトに対するイベントの種類に応じて異なる値が算出され、内面的成長度合いの値が更新される。AOサーバ13側で仮想ペットなどの性格を管理する場合、各クライアントから伝送されて来るメッセージの種類に応じて、例えば、チャットで話かけられる毎に成長パラメータの所定の指数が0.1ポイント加算され、「ほめるボタン」Dが押されてほめられる毎に0.2ポイント加算され、「しかるボタン」Fが押されて叱り付けられる毎に0.2ポイント減算される等の所定の加減剰余の成長計算式に基づいて算出される。

【0161】共有仮想空間内における仮想生命オブジェクトの自律的な挙動を管理するAOサーバ13に、各仮想生命オブジェクトの成長度合いを示す成長パラメータ管理テーブルを設け、各仮想生命オブジェクトの誕生日時を保持し、この時点からの経過時間に基づいて、各仮想生命オブジェクトの年齢に応じた成長パラメータを算出し、管理テーブルを更新する。

【0162】共有仮想空間内における複数の仮想生命オブジェクトの自律的な挙動を各々独立して管理するAOサーバ13（単一のAOサーバ13で複数の成長パラメータ管理テーブルを管理してもよく、複数のAOサーバ13、14で各々管理しても良い）を設け、それぞれの仮想生命オブジェクトの成長度合いを示す成長パラメータ管理テーブルを各々別個に設けて、各仮想生命オブジェクトの成長パラメータを独立して管理することができる。

【0163】その他、種々の応用例が考えられる。

【0164】例えば、共有仮想空間内における仮想生命オ

プロジェクトの自律的な挙動を管理するAOサーバ13に、各仮想生命オブジェクトにアクセスしたクライアントのID（ユーザが設定した仮想生命オブジェクトのニックネーム等）を管理する管理テーブル（成長管理テーブルに、そのような管理項目を追加してもよいし、独立した管理テーブルとしてもよい。以下同様）を設け、このIDのクライアントからのアクセスに応じて親密な感情表現を示すイベントを起動するようにしてもよい。このように、AOサーバ13側で親密度（アクセスの回数とその内容）を保持しておくことにより、飼い主がワールド（共有仮想空間）に入るとそばに近寄ってくるペット（仮想生命）オブジェクトを実現することができる。

【0165】仮想生命オブジェクトをセットまたはリセットしたクライアントのIDを、その仮想生命オブジェクトの飼い主として、成長パラメータ管理テーブルに格納し、仮想生命オブジェクトの寿命が尽きるまで、IDの更新を不可とすると共に、寿命が尽きた時点でIDを消去するようにしてもよい。これにより、仮想生命オブジェクトを誕生（セット）させた飼い主に忠誠を尽くす誠実型ペットを実現することができる。さらに、寿命が尽きた（リセットされた）時点で、自動的にその仮想生命オブジェクトの子供を誕生させ、その子供に同じ飼い主のIDを飼い主として初期設定させるようにしてもよい。これにより、その仮想ペットの子孫も先祖の飼い主になつくことになる。

【0166】仮想生命オブジェクトをセットまたはリセットしたクライアントの履歴を管理する履歴管理テーブルをAOサーバ13に設け、クライアントからのアクセス頻度に応じて、アクセス頻度の高いクライアント程、より親密な感情表現を示す挙動シーケンスを起動させることができる。逆に、アクセス頻度が少ないと、親密度が薄れ、アクセス頻度で親密度が変化する浮気型ペットが実現される。

【0167】共有仮想空間内でのクライアントの位置に対応して仮想生命オブジェクトの移動を制御すれば、共有仮想空間にクライアント（飼い主）が入ってきたとき、仮想生命オブジェクトがその目の前にすぐに現れ、飼い主についてまわる仮想生命オブジェクトを実現することができる。

【0168】共有仮想空間内でのイベント（例えば、他のクライアントから食事を与えられたという体験）を、仮想生命オブジェクトからテキストベースでのチャットウィンドウを介してクライアントへ体験報告させるようにすることで、親密な感情表現を行うことができる。

【0169】共有仮想空間内でのイベントを、仮想生命オブジェクトから音声チャットを介してクライアントへ体験報告することで、親密な感情表現を行うようにしてもよい。テキスト読み上げソフトによるテキスト/音声変換によりアナログ音声メッセージとして音声チャット機能を介して通知してもよく、また、予め何種類かの音

声メッセージをサンプリングし、圧縮してデジタル音声圧縮データとしてHDD（AOサーバ13側またはクライアントPC1側のいずれでもよい）に格納しておいて、複数の音声メッセージの中から適切な音声メッセージを選択的にHDDから読み出して、伸張した後、音声チャット機能を介してアナログ音声メッセージとして通知するようにしてもよい。このようなテキストと音声の変換については、例えば特開平7-105848号公報に開示されている技術を用いることができる。

【0170】次に、既存の通信インフラストラクチャを利用した飼い主への通知機能や、既存の通信インフラストラクチャを利用した仮想ペットへの簡易的な操作機能について説明する。

【0171】例えば、仮想生命オブジェクトの食欲指数が高まった場合や、寿命残時間が所定値以下になった時点で、仮想生命オブジェクト（AOサーバ13）は、成長パラメータ管理テーブルに予め設定されている連絡手段（図8）を用いて、飼い主の連絡先へ、その状態が変化した旨を、メールサーバ15またはコミュニケーションサーバ16を介して通知する。これにより、飼い主は、屋外へ外出中であるとか、パーソナルコンピュータがうまく動作しないなど、即座にパーソナルコンピュータにアクセスできない状況下であっても、自分が育てている仮想ペットの要求を確認し、コミュニケーションを図ることが可能となる。

【0172】一方、この種の仮想ペットを飼育するサービスを提供することによって、利益を得るサービスプロバイダにとっては、ユーザに対して、そろそろアクセスしなければならぬと言う一種の動機付けを喚起することができ、定常的なアクセスを確保することで、より健全な経営状態を維持することが可能となるといった副次的な効果も得られる。

【0173】既存の通信インフラストラクチャを利用した飼い主への通知機能を実現するための具体例は以下の通りである。

【0174】共有仮想空間における仮想生命オブジェクトの自律的な挙動を管理するサーバのうち、コミュニケーションサーバ16に、各仮想生命オブジェクトの飼い主である各ユーザへのメッセージ通知手段（通知方法）の種類と、その通知先を管理するためのコミュニケーション管理テーブル（図8の成長パラメータ管理テーブルの飼い主データに対応するテーブル）を設け、仮想生命オブジェクトの状態遷移（AOサーバ13からコミュニケーションサーバ16に通知される）に応じて、コミュニケーション管理テーブルに予め登録されている通知手段を用いて、その通知先へメッセージを送信する。これにより、共有仮想世界から、現実世界に実際に連絡が行われる。

【0175】このコミュニケーション管理テーブルで、各仮想生命オブジェクトの飼い主である各ユーザを特定

するためのIDと、そのIDのユーザからのアクセス履歴を管理し、各ユーザからのアクセス履歴に基づいて、一定期間以上アクセスのないユーザに対して、例えば、「近ごろ、会いに来てくれないので、さみしい....」などのメッセージを送信することができる。

【0176】さらに、各ユーザからのアクセス履歴に基づいて、複数のメッセージ文例の中から適切なメッセージ文を選択し、この選択したメッセージを送信するようにしてもよい。ユーザからのアクセスがあった翌日には、仮想生命オブジェクトからユーザに、「昨日は、遊んでくれてどうもありがとう。また遊ぼうね。」などのメッセージを送り、1週間アクセスがないと、「忙しいですか？たまには、遊びに来て下さい。」などのメッセージを送る。

【0177】仮想生命オブジェクトの状態遷移を、成長パラメータ管理テーブルの更新内容に基づいて検出し、複数のメッセージ文例の中から適切なメッセージ文を選択し、この選択したメッセージを送信することができる。例えば、食事が沢山与えられた翌日には、「まだ、おなかがいっぱいだよ〜。」などのメッセージを送り、1週間アクセスがないと、「おなかが空いて死にそうだよ〜。」などのメッセージを送る。1週間で1歳年を取るシナリオとした場合は、毎週、誕生日が来て年を取る毎に、「私は今日で10歳になりました。成長した姿をみに来て下さい。」などのメッセージを送り、寿命が残り少なくなると、「そろそろ遠い所へ旅立たなければなりません、最後を見とどけに来てください。」などのメッセージを送る。

【0178】コミュニケーション管理テーブルのうち、電子メールに関する部分のテーブルをメールサーバ15に設け、メールサーバ15のコミュニケーション管理テーブルで、各仮想生命オブジェクトの飼い主である各ユーザの電子メールアドレス（ユーザを特定するIDも兼ねる）に対してテキストによるメッセージを送信することができる。

【0179】インターネット7を経由する電子メール以外に、PIAFS(PHS Internet Access Forum Standard)方式や、αDATA32方式対応PHSで電子メールサービスを受けられるPHS端末23や、ページャ（ポケットベル端末24）などに対して、コミュニケーションサーバ16から公衆電話回線網17を介してメッセージを通知し、それらのLCD上にメッセージを表示させるようにしてもよい。

【0180】あるいは、コミュニケーション管理テーブルで、各仮想生命オブジェクトの飼い主である各ユーザの通知先電話番号（ユーザを特定するIDも兼ねる）を管理し、コミュニケーションサーバ16からこの電話番号に対して自動発信して、音声によるメッセージを送信することもできる。この場合、テキスト読み上げソフトによるテキスト/音声変換によりアナログ音声メッセー

ジとして通常の電話機8や携帯電話機（PHS端末23）を介して通知してもよく、また、予め何種類かの音声メッセージをサンプリングし、圧縮してデジタル音声圧縮データとしてHDDに格納しておいて、複数の音声メッセージの中から適切な音声メッセージを選択的にHDDから読み出して、伸張した後、アナログ音声メッセージとして通知するようにしてもよい。

【0181】コミュニケーション管理テーブルで、各仮想生命オブジェクトの飼い主である各ユーザの通知先ファクシミリ電話番号（ユーザを特定するIDも兼ねる）を管理し、コミュニケーションサーバ16からこの電話番号に対して自動発信して、ファクシミリ19にメッセージを送信してもよい。

【0182】さらに、上述した仮想ペットからのメッセージの通知を受けた電話機18等の端末を用いて、この端末から、自分の仮想ペットに対して、簡易的な操作を行なうことで、双方向のコミュニケーションを図ることが可能となる。

【0183】既存の通信インフラストラクチャを利用して、仮想ペットへの簡易的な操作機能を実現するための具体例は以下の通りである。

【0184】例えば、共有サーバ12は、ユーザからPHS端末23、電話機18などを介してユーザを特定するIDを伴って（Caller ID等）返送されてくる操作コマンドを解釈し、対応する仮想生命オブジェクトへのメッセージとして認識し、AOサーバ13に通知して、仮想生命オブジェクトの状態遷移（成長パラメータの更新）へ反映させることができる。

【0185】この返信されてくる操作コマンドとしては、例えば、電話機18のプッシュボタンが操作されることで発信されるDTMF(Dual-tone Multifrequency)信号を利用することができる。

【0186】この他、例えばデネットフォニックコミュニケーションズ社の音声ブラウザWeb-On-Callを用いて、仮想ペットに対して各種の通知を行うようにすることもできる。この音声ブラウザによれば、通常の電話機18から音声によってサーバにアクセスすることができる。また、この音声ブラウザは、テキスト読み上げ機能を有しているので、仮想ペットからのメールを音声で聴くことも可能である。さらに、テキストをファックスや電子メールで送ることもできる。

【0187】なお、共有サーバ12、AOサーバ13、14、メールサーバ15、およびコミュニケーションサーバ16は、各種の機能を分担し、全体的なシステムとして、各クライアントPCに共有仮想空間を提供するサービスを実現している。

【0188】次に、この共有仮想空間におけるチャットについて説明する。チャットには、パブリックチャットとパーソナルチャットがある。パブリックチャットにおいては、共有仮想空間内の所定の位置に位置する1人の

クライアント（ユーザ）が発生したチャットの内容が、近傍の他のクライアント（他のユーザ）へ伝達される。

【0189】これに対して、パーソナルチャットにおいては、対象とされる相手が指定される。この指定操作は、例えば、所定の仮想生命オブジェクトをキーボード42のシフトキーを押しながらマウスのボタンをクリックすることで行われる。このように、相手が指定された場合には、そのクリック操作を行ったクライアントが発生するチャットが、その指定した仮想生命オブジェクトのユーザに対してのみ伝送される。

【0190】チャットの内容は、ボイスチャットの場合、音声信号により伝送され、テキストチャットの場合、テキストにより伝送される。ボイスチャットの場合は、マイクロホン36を介して取り込んだ音声データが、近傍または指定した仮想生命オブジェクトのクライアント（ユーザ）に伝送され、そのクライアントPCの有するスピーカ37、38から出力される。

【0191】これに対して、テキストチャットの場合は、キーボード42を操作することで入力されたテキストが、やはり近傍または指定した仮想生命オブジェクトのクライアントのクライアントPCに伝送される。そして、そのテキストは、そのクライアントPCのCRTモニタ45上に表示される。

【0192】ボイスチャットとテキストチャットのいずれを行うかは、ユーザが選択することができる。

【0193】次に、クライアントPCのCRTモニタに表示される具体的な表示例について説明する。図17は、例えばクライアントPC1から共有サーバ12にアクセスした場合のCRTモニタ45上の表示例を表している。この表示例においては、「Community Place Bureauに接続しました」のメッセージが3次元共有仮想空間の画像に重畳表示されている。なお、この画面の右側には、チャットの内容を表示する領域「Chat Log:」と、各種のアクションボタンが表示されている。

【0194】ユーザは、自分が共有仮想空間内において飼育しようとするペットを選択するとき、表示画面中の「MultiUser」の項目をクリックする。このとき、図18に示すように、メニューバーが表示される。ユーザは、この中から「Select Pet」を選択する。すると、図19に示すように、「View Pet」のウインドウが表示され、このウインドウに各種のペットの画像が表示される。ユーザは、このペットの中から所望のペットを選択する。なお、このとき表示されるペットの画像は、初期状態の画像、すなわち、誕生するときの画像とされる。上述したように、このペットの画像は、成長に伴って徐々に変化していく。そして、選択が完了したとき、ユーザは、「SELECT」のボタンをマウス41を操作してオンする。

【0195】このようにして、ペットの選択が完了すると、図20に示すように、ペットに関する情報を登録す

る「Registration」のウインドウが表示される。ユーザは、ここで、ペットの名称「Pet nickname」、自分自身（飼主）の名称「Owner name」、およびアドレスと電話番号「Address/TelephoneNo.」をキーボード42を操作して入力する。

【0196】さらに、上述した連絡手段または通知手段としてのコミュニケーションツール「Communication tool」を、E-mail, Telephone, PHS-mail, Facsimile, Pocketbellの中から、選択、指定する。

【0197】図20は、このようにして、仮想ペットとして、例えば猿が選択された場合の表示例を表している。この仮想ペット（猿）は、上述したように、各種のイベントに対応して更新される成長パラメータに基づいて成長する。図21は、図20に示した場合より成長した状態を表している。図20に示す場合より図21に示す場合の方が、身体が全体的に大きくなっており、顔もより大人の顔に変化している。

【0198】このようにして、より現実に近い変化を共有仮想空間において楽しむことができる。

【0199】次に、具体的な表示例を参照しながら、仮想ペットが、障害物との衝突を検出することなく、実世界では有り得ない行動を回避する方法を説明する。図23は、例えば、クライアントPC1から共有サーバ12にアクセスした場合に提供される3次元共有仮想空間の鳥瞰的な表示例を表している。図24は、図23の3次元共有仮想空間の一部を側面上方から見下した画像が表示されている。

【0200】実世界では有り得ない行動とは、障害物、例えば、図24の家A1の壁A2に衝突して、家A1の壁A2の中に入り込むなどの行動であり、実世界では、ペットが家A1の壁A2に衝突すると、そこで立ち止まるか、または、家A1の壁A2に入り込まないように移動する。

【0201】図25は、図23の3次元共有仮想空間の他の一部を斜め上方から見下した画像が表示されている。この例の場合、広場B1が段Cの高さ分だけ、広場B2より高い位置にあり、位置E1乃至E9を曲がり角または分岐点とする通路Dが表示されている。実世界では有り得ない行動は、この例の場合、例えば、仮想ペットが、広場B2から広場B1に向かって歩いているとすると、仮想ペットが、段Cを登らずに、段Cに衝突し、広場B1の下に潜り込んでしまう、つまり、土の中に入り込むような行動である。実世界では、仮想ペットは、段Cの所に来ると、段Cを登り、広場B1にたどり着く。

【0202】本発明を適用した共有仮想空間提供装置では、このような、3次元共有仮想空間において、実世界では有り得ない行動を回避するに、仮想ペットが移動できる範囲を予め設定し、仮想ペットの行動に制限を加える。

【0203】図25の通路Dは、仮想ペットが移動でき

る範囲を表している。下記に、通路Dを3次元共有仮想空間に設定するプログラムの一部を示す。なお、このプログラムはJavaのスクリプトで作成されている。

【0204】

```
1: public class PathNode
2: {
3:     float m_x,m_y,m_z;
4:     short m_type;
5:     Vector m_link;
6: }
7: public class Pathlink
8: {
9:     float m_width;
10:    short m_type;
11:    PathNode[] m_node;
12:    boolean isOnTheRoad(float x,float z);
13:    float calcHeight(float x,float z);
14:}
```

【0205】図26は、図25の表示例を真上から見た場合の画像の簡略図(図26(A))とパスリンクE2E7から真横に見た場合の画像の簡略図(図26

(B))を表している。次に、図26を参照して上記の仮想ペットの移動範囲を設定するプログラムの内容を説明する。なお、位置ノードN1乃至位置ノードN9は、図25の位置E1乃至E9に対応する位置を表すノードである。パスリンクは、ここでは、位置ノードと位置ノードを結んだ線で、共有仮想空間において、通路に相当する。次に、図26を参照して上記プログラムの内容を説明する。

【0206】公開クラス宣言されたクラス名PathNode(1行目乃至6行目)は、図26の位置ノードN1乃至位置ノードN9を設定するクラスである。2行目で、位置ノードN1乃至位置ノードN9のX軸、Y軸、およびZ軸の座標が設定される。なお、いまの場合、図26に示すように、X軸方向はパスリンクN4N5と平行な方向、Z軸方向は、パスリンクN2N4と平行な方向、およびY軸方向は、X軸とY軸に対して垂直な方向である。3行目で、位置ノードの種類が設定され、4行目で、位置ノードを結ぶことによりできるパスリンクを、Vector形式(リスト形式)で保持する。

【0207】次に、公開クラス宣言されたクラス名PahtLine(7行目乃至14行目)では、パスリンクN1N2、パスリンクN2N3、パスリンクN3N4、パスリンクN4N5、パスリンクN5N6、パスリンクN6N7、パスリンクN7N1、パスリンクN1N8、パスリンクN8N9、パスリンクN9N6、パスリンクN3N8の条件を設定するクラスである。9行目で、通路とされたパスリンクの幅や広場とされたパスリンクの直径を指定し、10行目で、パスリンクの種類を指定している。例えば、ここで、パスリンクを通路また、広場とすることができる。11行目

で、位置ノードの配列を管理している。

【0208】次に、12行目で、位置ノードのX軸座標とZ軸座標の値がパスリンク上にあるかどうかを判定し、13行目で、位置ノードのY軸座標(高さ)をX軸座標とZ軸座標から算出する。

【0209】上述したような、仮想ペットが移動できる範囲(例えば、図25の通路D)における仮想ペットの自律行動について、図27のフローチャートを参照して説明する。

【0210】はじめに、ステップS21において、仮想ペットが移動できる範囲を規定するデータが、例えば、HHD13aの中の移動範囲管理テーブルに登録される。次に、ステップS22において、仮想ペットに対応するアバタの位置が検出され、ステップS23においては、検出されたアバタの位置に基づいて、仮想ペットの位置が発生される。仮想ペットは、所定の距離を保ちながら、アバタに追従するようになされている。

【0211】次に、ステップS24において、ステップS23で発生した仮想ペットの位置が、仮想ペットが移動できる規制範囲内であるか否かが判定される。ステップS23で発生した仮想ペットの位置が、規制範囲外であると判定された場合、ステップS25に進み、発生した仮想ペットの位置に最も近い規制範囲内の位置が検索される。

【0212】次に、ステップS26において、仮想ペットの位置が、ステップS25で検索された位置に修正される。次に、ステップS27において、仮想ペットは、さらに、アバタに追従して移動する。

【0213】ステップS24において、ステップS23で発生した仮想ペットの位置が、仮想ペットが移動できる規制範囲内であると判定された場合、ステップS27に進み、仮想ペットは、さらに、アバタに追従して移動する。

【0214】このように、3次元共有仮想空間に、予め仮想ペットが移動できる範囲を設定し、仮想ペットの行動を制御することによって、例えば、仮想ペットが障害物との衝突を検知する必要なく、現世界では有り得ない行動を回避することができる。

【0215】また、図28に示すように、仮想ペットが、移動範囲内で、自由に移動できるように、ランダムな数値を、移動範囲管理テーブルに対応して管理する、例えば、乱数管理テーブル(図示せぬ)を設け、仮想ペットが、乱数管理テーブルのデータに基づいて、移動範囲内をランダムに移動できるようにすることもできる。

【0216】なお、いまの場合、仮想ペットの移動範囲を設定するプログラムの例を説明したが、実際には、通路情報のセットアップルーチン、設定された座標が通路の上にあるかを判定するルーチン、通路上の任意の座標における高さを計算するルーチンなどのプログラムが存在する。

【0217】

【発明の効果】請求項 1 に記載の共有仮想空間提供装置、請求項 2 に記載の共有仮想空間提供方法、および請求項 3 に記載の提供媒体によれば、共有仮想空間内に存在する仮想生命オブジェクトの移動範囲を規定するデータを記憶し、仮想生命オブジェクトの共有仮想空間における新たな位置のデータを発生し、発生された位置を規定されている移動範囲内の位置に修正し、修正された位置に仮想生命オブジェクトが移動するようにしたので、仮想ペットが、障害物との衝突を検知する必要なく、現実世界ではあり得ない行動を回避することが可能となる。

【図面の簡単な説明】

【図 1】センサー、イベント、ルーティング、およびスクリプトの関係を説明する図である。

【図 2】ルーティングを説明する図である。

【図 3】本発明を適用した共有仮想空間提供システムの構成例を示すブロック図である。

【図 4】図 1 のクライアント PC 1 の構成例を示すブロック図である。

【図 5】図 3 のシステムの動作を説明するディスプレイの写真である。

【図 6】図 3 のシステムの動作を説明するディスプレイの写真である。

【図 7】図 3 のシステムの動作を説明するディスプレイの写真である。

【図 8】成長パラメータ管理テーブルの例を示す図である。

【図 9】アクションパネルを説明するディスプレイの写真である。

【図 10】成長パラメータの授受を説明する図である。

【図 11】3D オブジェクトを構成する仮想生命オブジェクトのノードを説明する図である。

【図 12】図 11 に示したノードに対応する表示例を示すディスプレイの写真である。

【図 13】仮想生命オブジェクトの体格指数を説明する図である。

【図 14】仮想生命オブジェクトの機嫌指数を説明する

図である。

【図 15】仮想生命オブジェクトの活発性指数を説明する図である。

【図 16】仮想生命オブジェクトの知能指数を説明する図である。

【図 17】共有仮想空間のディスプレイの表示例を示す写真である。

【図 18】共有仮想空間のディスプレイの表示例を示す写真である。

【図 19】共有仮想空間のディスプレイの表示例を示す写真である。

【図 20】共有仮想空間のディスプレイの表示例を示す写真である。

【図 21】共有仮想空間のディスプレイの表示例を示す写真である。

【図 22】共有仮想空間のディスプレイの表示例を示す写真である。

【図 23】共有仮想空間のディスプレイの表示例を示す写真である。

【図 24】共有仮想空間のディスプレイの表示例を示す写真である。

【図 25】共有仮想空間のディスプレイの表示例を示す写真である。

【図 26】図 25 の表示例を真上から見た画像の簡略図と、断面図である。

【図 27】規制範囲内における仮想ペットの自律行動について説明するフローチャートである。

【図 28】図 25 の通路 D 上を徘徊する仮想ペットの軌跡を示す図である。

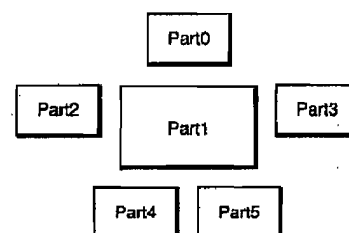
【符号の説明】

1 乃至 3 クライアント PC, 7 インターネット, 10 WWWサーバ, 12 共有サーバ, 13, 14 AOサーバ, 15 メールサーバ, 16 コミュニケーションサーバ, 18 電話機, 19 ファクシミリ, 23 PHS 端末, 24 ポケットベル端末, 30 CPU, 31 ハードディスク, 39 モデム, 45 CRT モニタ, 46 RAM

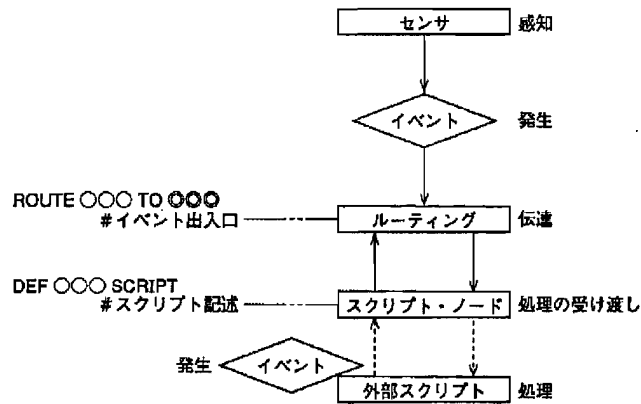
【図 2】

ROUTE Node Name.event Out Name TO Node Name.event In Name
(ルート宣言) (ノード名) (イベント名) (ノード名) (イベント名)

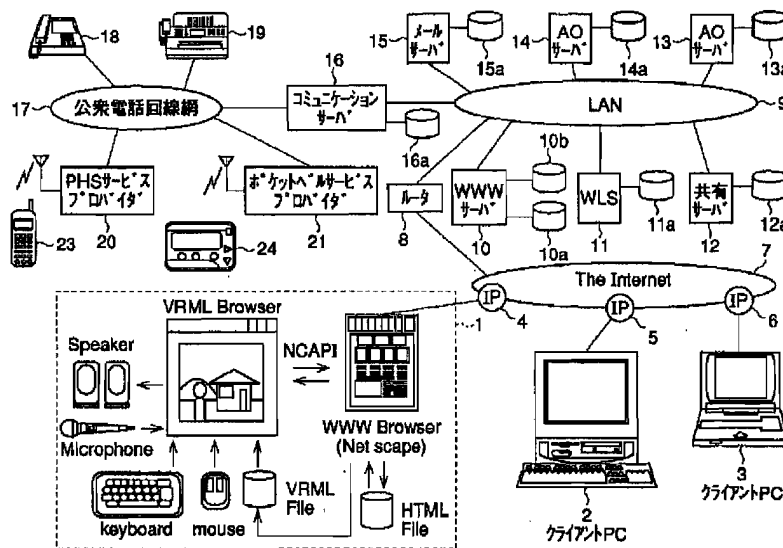
【図 11】



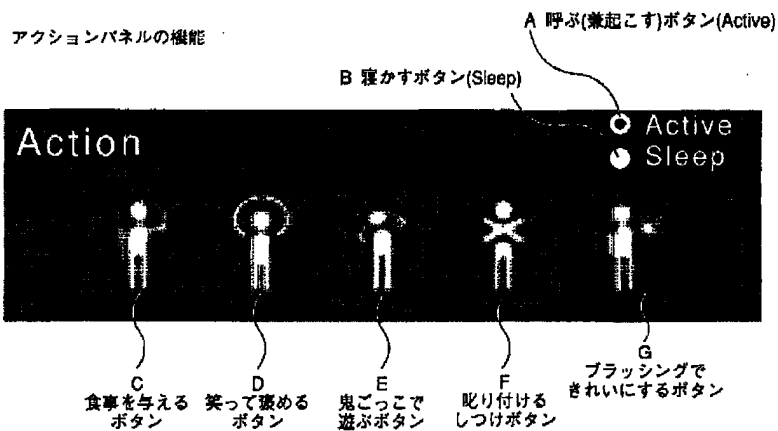
【図1】



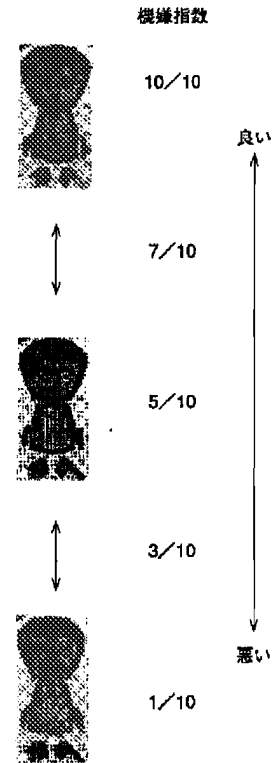
【図3】



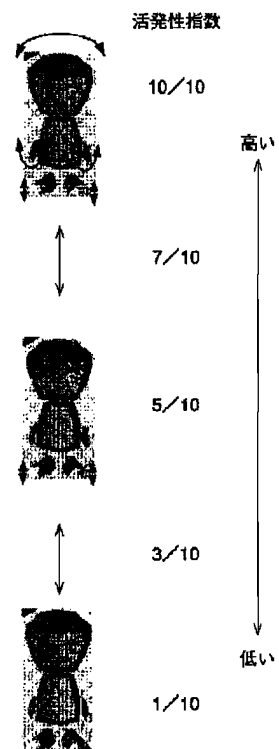
【図9】



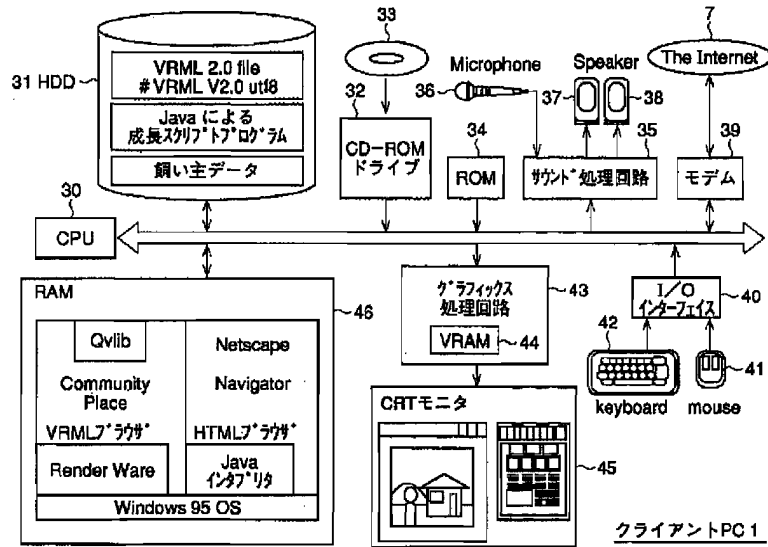
【図14】



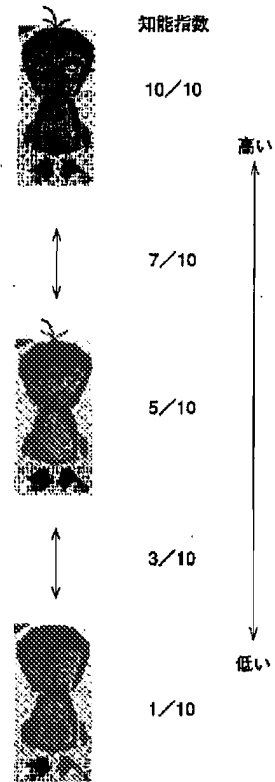
【図15】



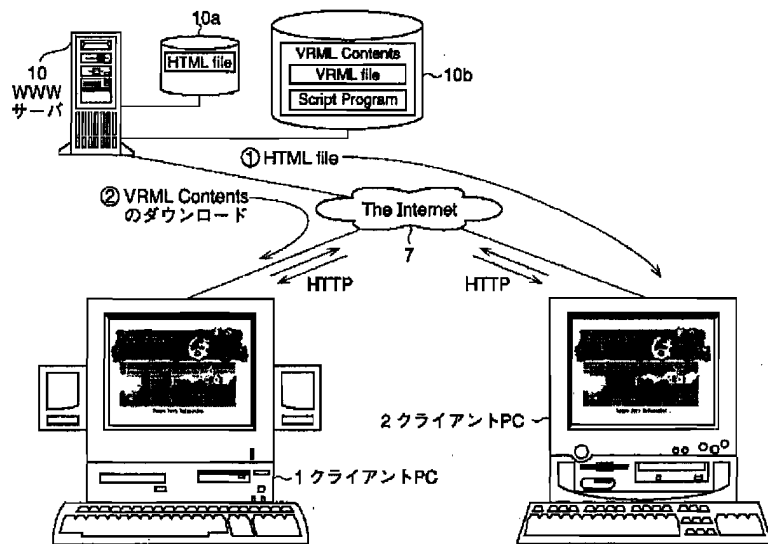
【図4】



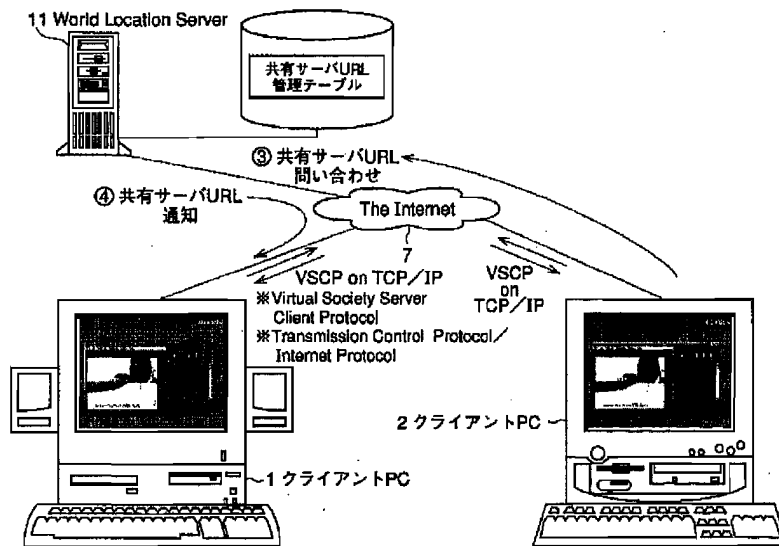
【図16】



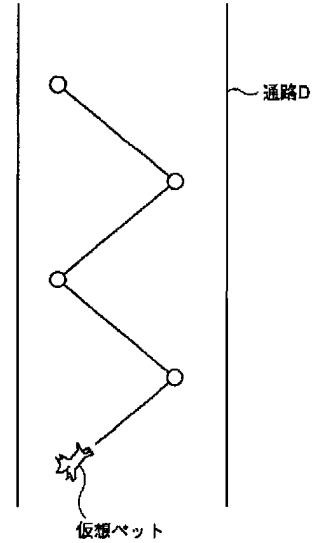
【図5】



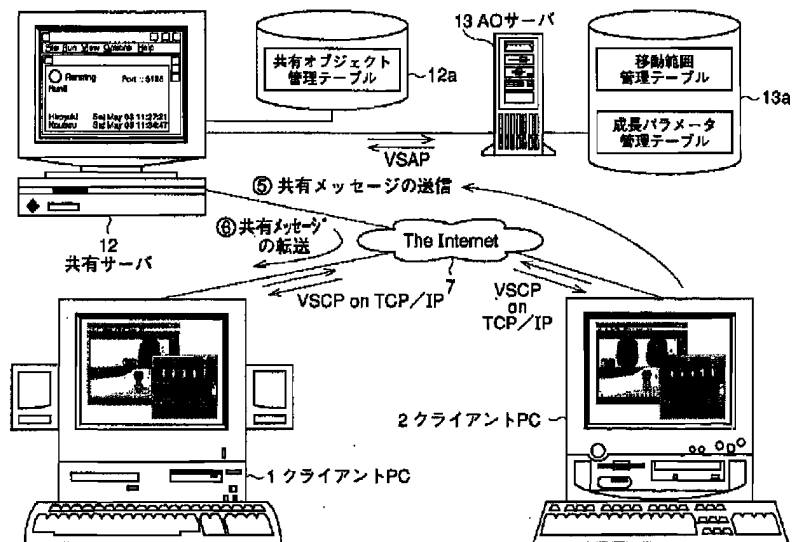
【図6】



【図28】



【図7】



【図8】

成長パラメータ管理データベース											
No	飼い主データ				成長パラメータ				連絡先		リザーブ
	3D オブジェクト ID	座標軸	種類	性別	ニックネーム	誕生日時	本籍 ワールド	成長パラメータ フィジカル パラメータ	メンタル パラメータ	連絡 手段	
1	Obj1	x1,y1,z1	さる	雄	Taro	970112	Zoo			0 hanaya@lpd.sony.co.jp	
2	Obj1	x2,y2,z2	ねこ	雌	Mike	970210	Park			1 03-5448-4443	

知能			
知能 指数	2/10	5/10	5/10

言語能力			
言語能力 指数	2/10	5/10	5/10

社交性			
社交性 指数	2/10	5/10	5/10

自主性			
自主性 指数	1/10	5/10	5/10

健康			
健康 指数	2/10	5/10	5/10

体格			
体格 指数	5/10	6/10	4/10

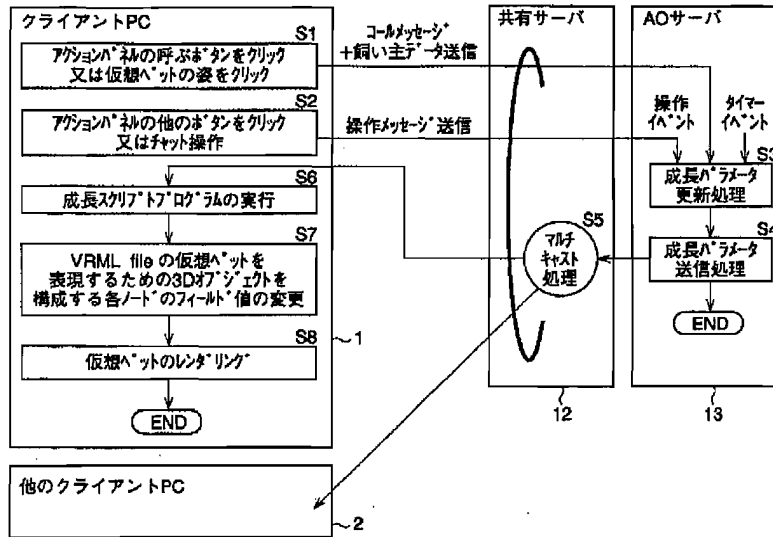
食欲			
食欲 指数	5/10	6/10	4/10

健康度			
健康度 指数	5/10	6/10	4/10

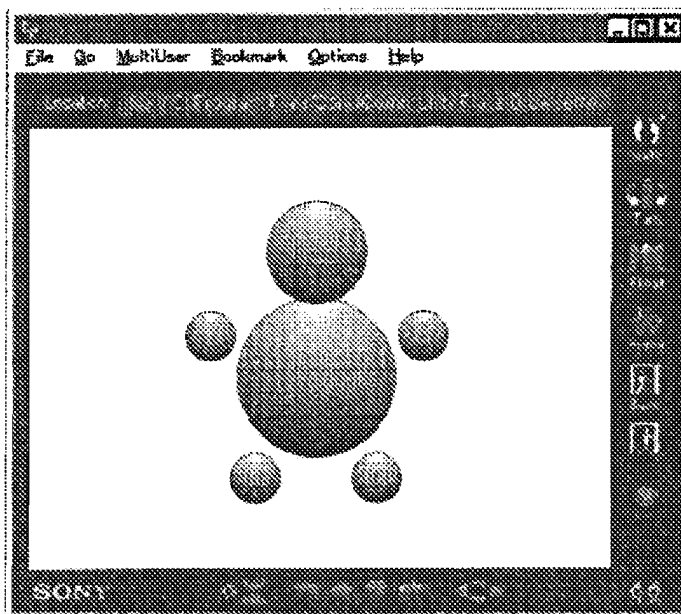
寿命			
寿命 時間	50	10	240

連絡手段			
連絡手段	0	電子メール	
	1	電話(アナログ音声)	
	2	PHSメールサービス	
	3	Fax	
	4	ポケットベル	
	5	リザーブ	
	6	リザーブ	
	7	リザーブ	

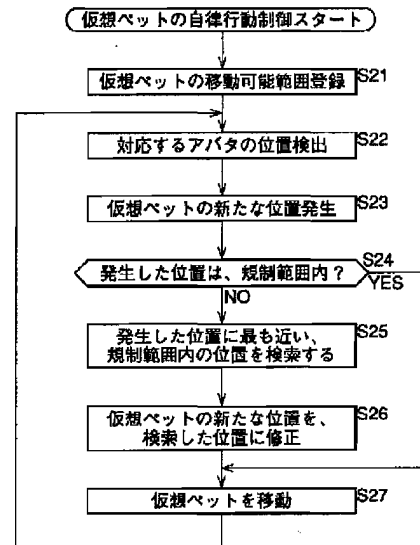
【図10】



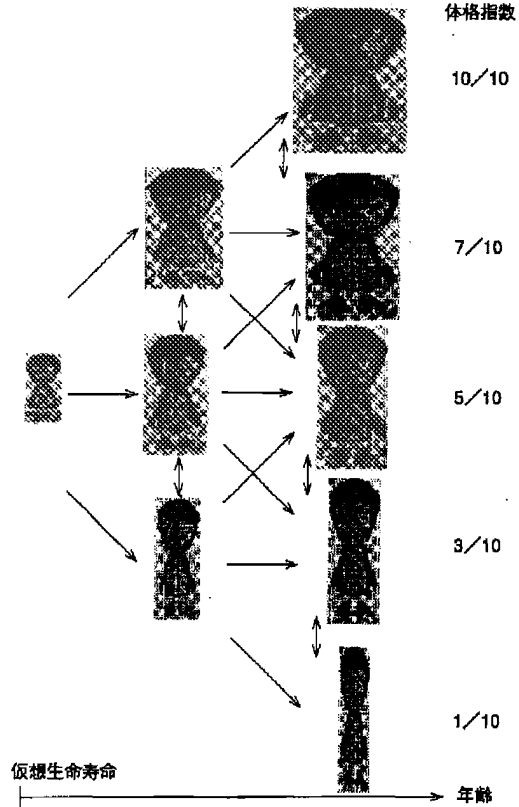
【図12】



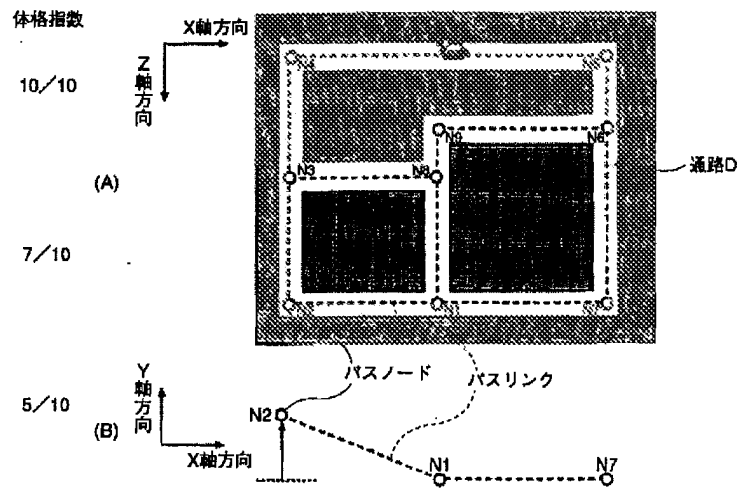
【図27】



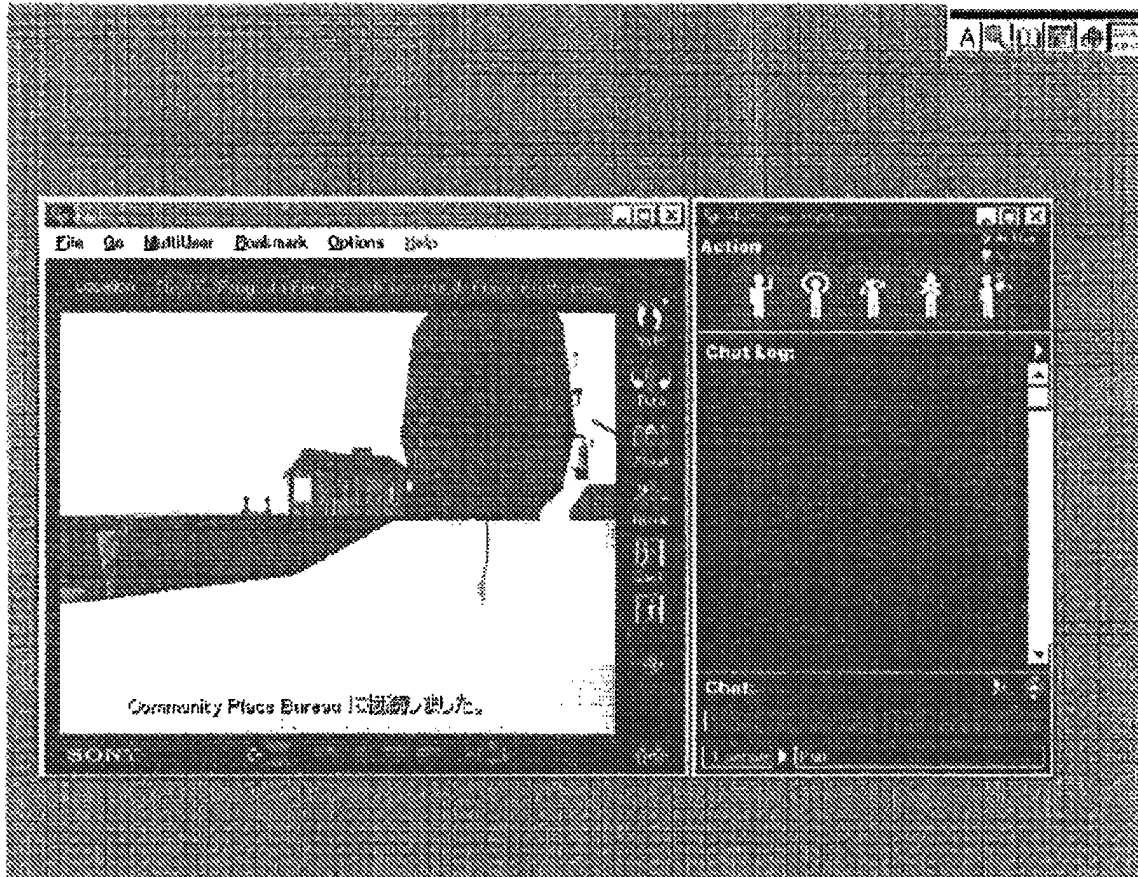
【図 13】



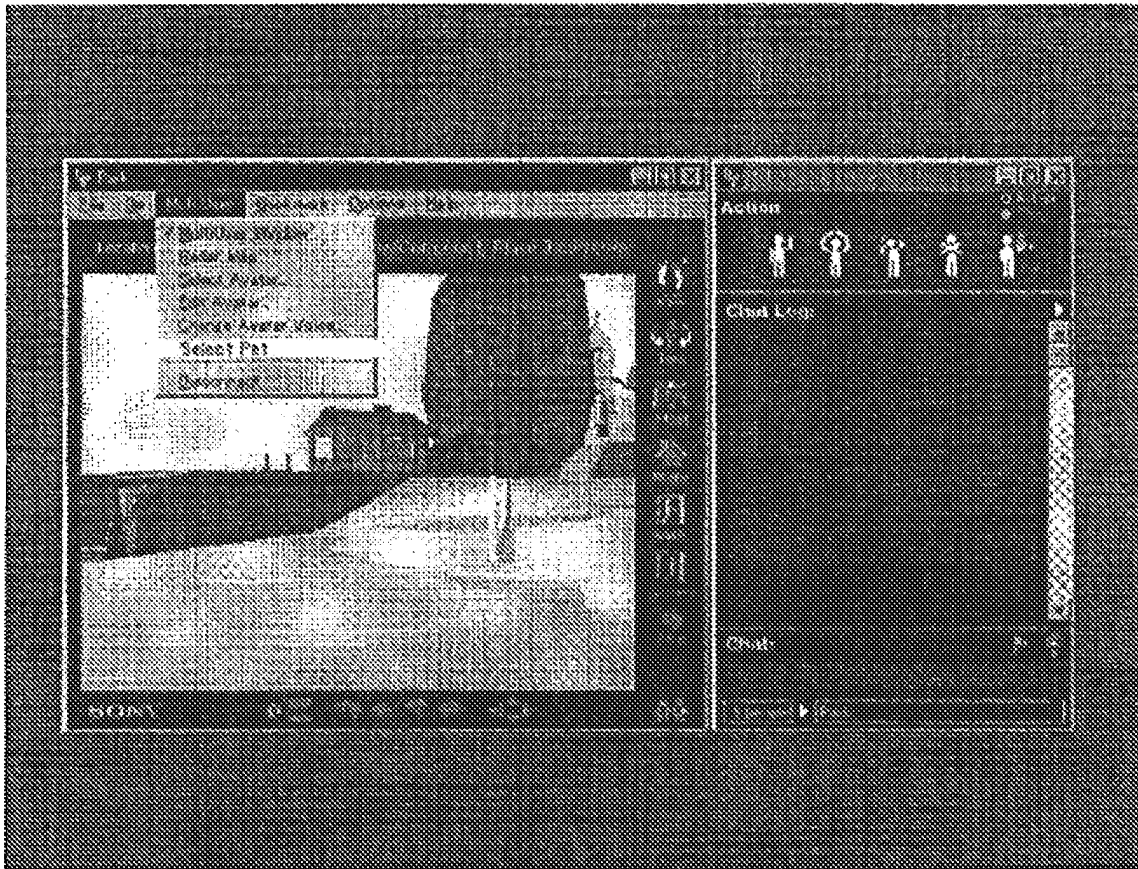
【図 26】



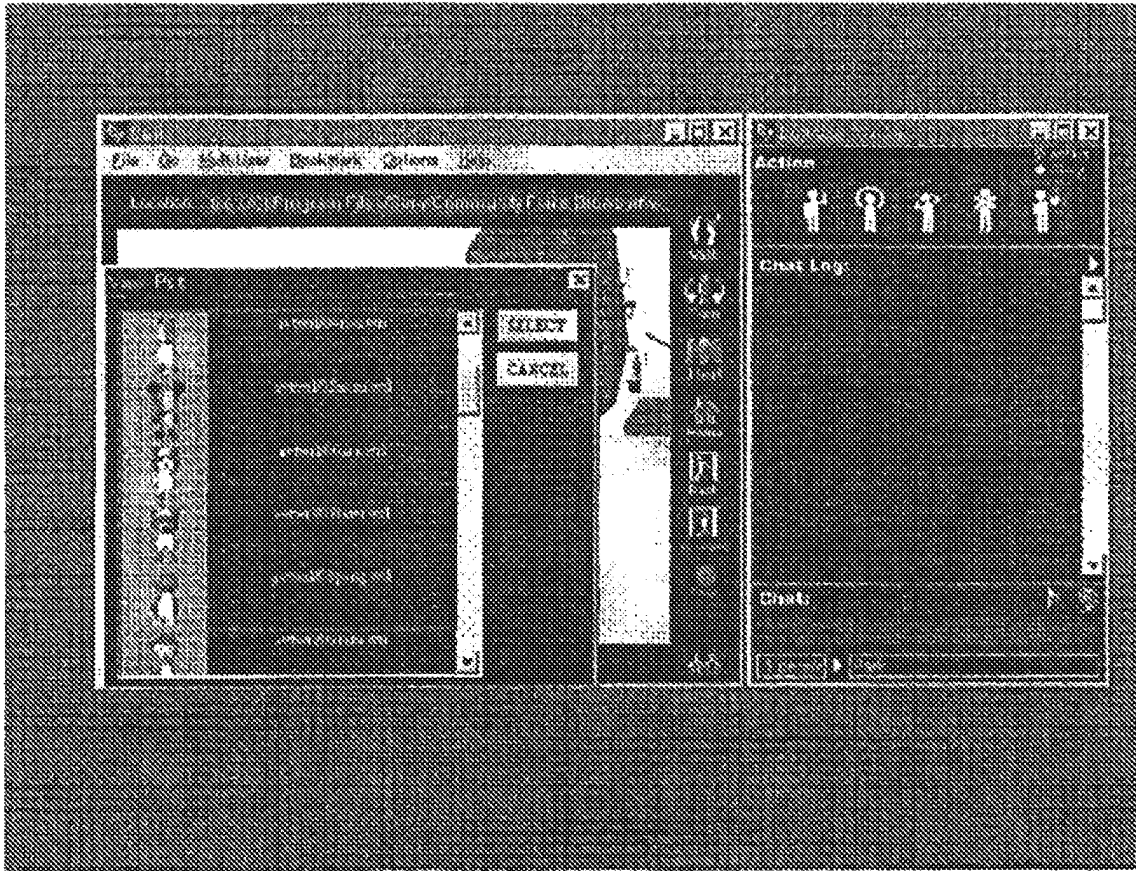
【図17】



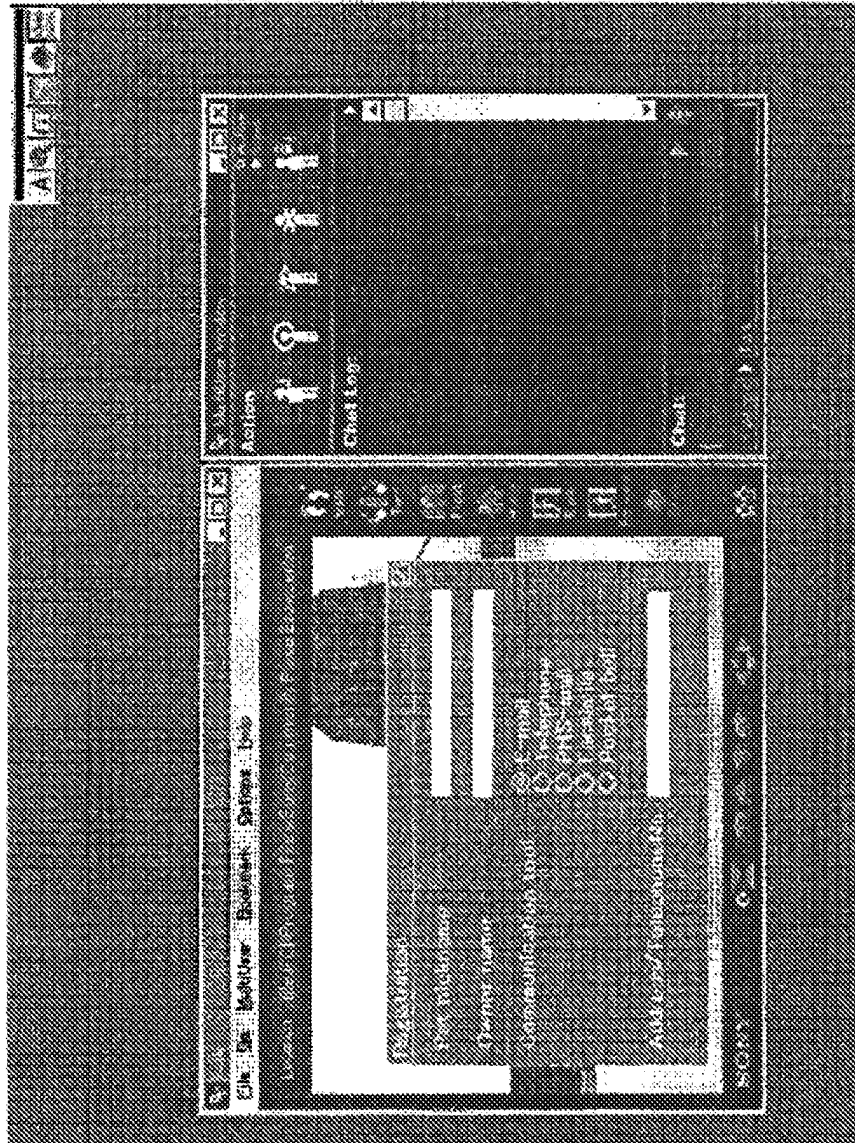
【図18】



【図19】



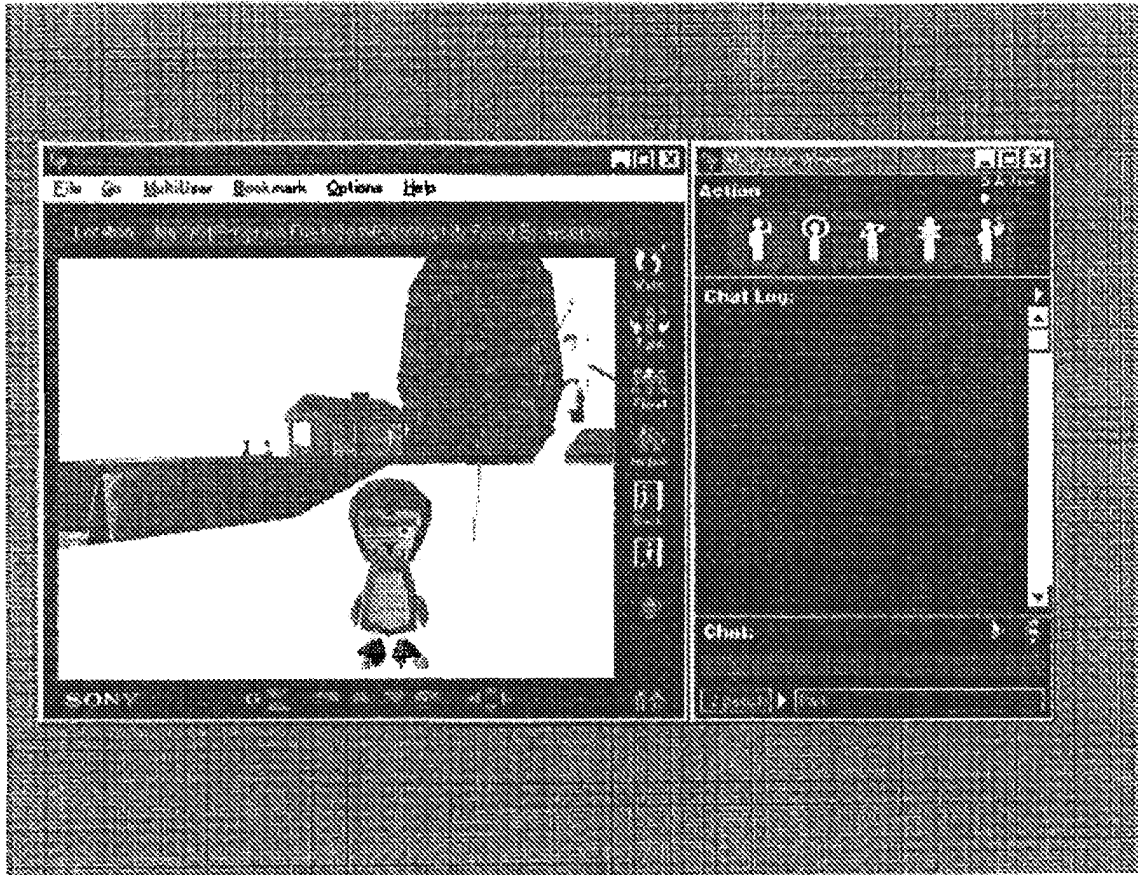
【図20】



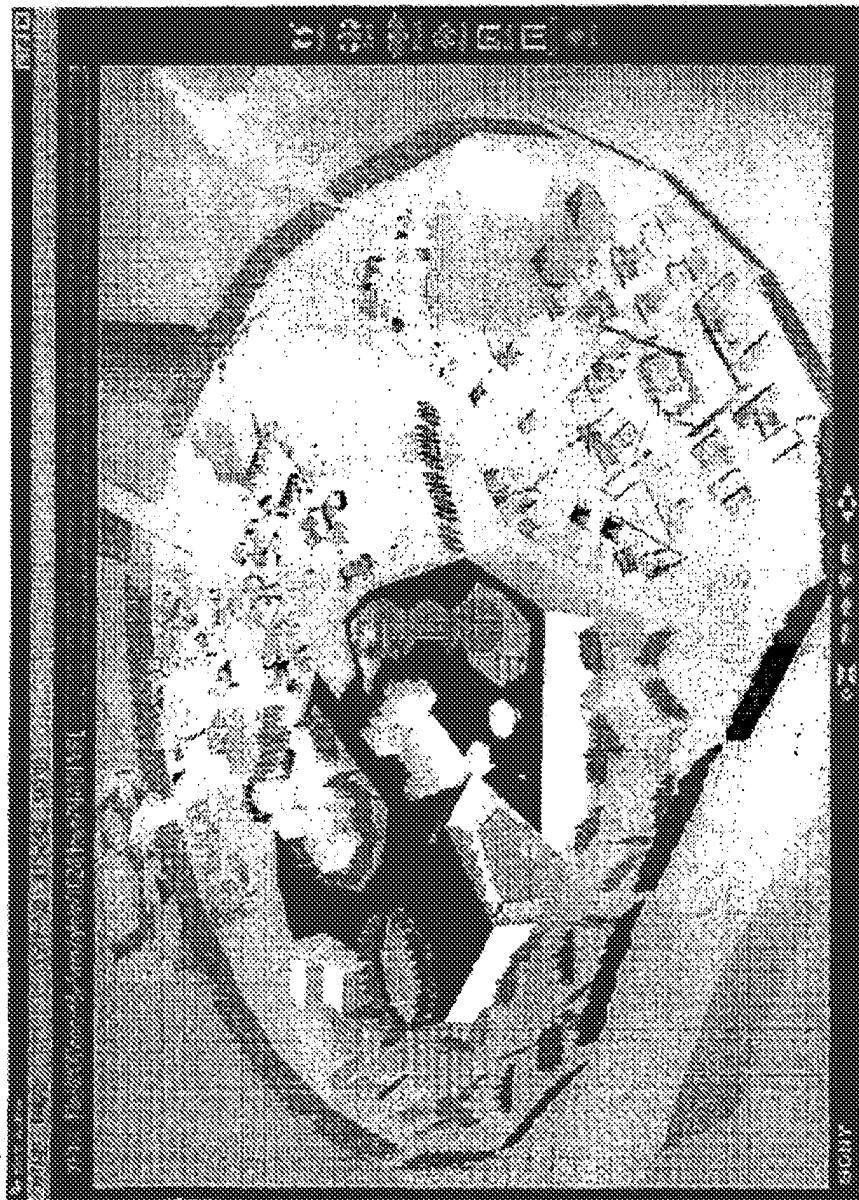
【図21】



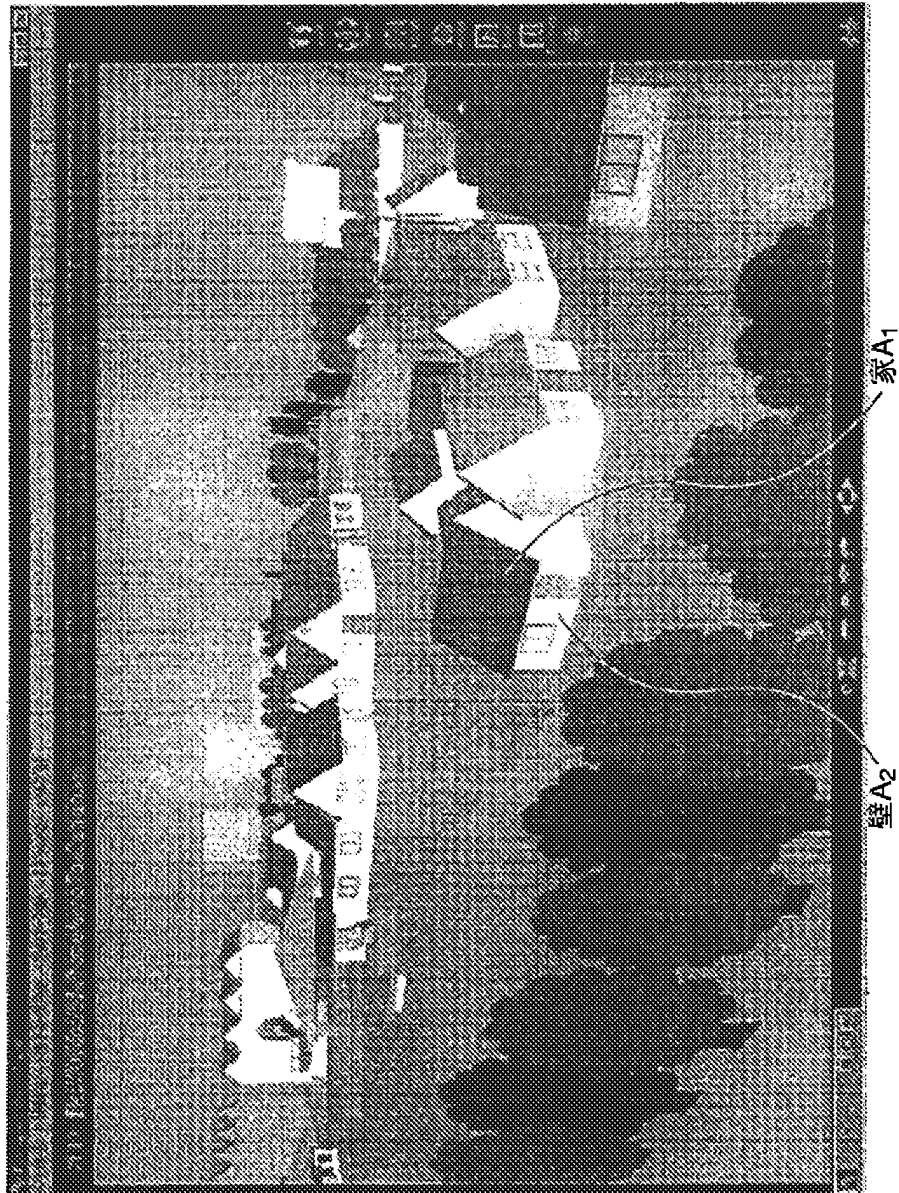
【図22】



【図23】



【図24】



【図25】

